Debian 13 aka. Trixie

Fixes

- Stromverbrauch im Schlafmodus mit einer Samsung 990 Evo NVMe

```
sudo su
nano /etc/default/grub
# Zeile die mit [GRUB_CMDLINE_LINUX_DEFAULT=] beginnt finden und
[nvme.noacpi=1] zu den bestehenden Parametern hinzufügen
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash nvme.noacpi=1"
#
update-grub
```

Firmware Updates

Um die Firmware einzelner Komponenten wie z.B. der SSD oder der Docking Station zu aktualisieren gibt es den Firmware Update Manager `fwupdmgr`.
Mit den folgenden Kommandos fürht man ein Update durch:

```
# Listet alle erkannten Geräte auf.
sudo fwupdmgr get-devices

# Aktualisiert die Datenbank via Linux Vendor Firmware Service
sudo fwupdmgr refresh

# Verfügbare Updates anzeigen
sudo fwupdmgr get-updates

# Updates herunterladen und Firmware aktualisieren
sudo fwupdmgr update

# Dienst stoppen oder starten
sudo service fwupd [stop|start]
```

System-Tools

.desktop

Quellen:

- https://wiki.ubuntuusers.de/.desktop-Dateien/
- https://specifications.freedesktop.org/desktop-entry-spec/latest/example.html



Restart des laufenden Desktops:
Bei GNOME geht das mit den Tasten Alt + F2 , und dann ein kleines R eingeben und mit ⏎ abschließen.

Beispiel `*.desktop` Datei:

org.freecad.FreeCAD.desktop

```
[Desktop Entry]
Name=FreeCAD
Comment=Feature-basierter parametrischer Modellierer
GenericName=CAD Application
Exec=/usr/bin/flatpak run --branch=stable --arch=x86_64 --
command=FreeCAD --file-forwarding org.freecad.FreeCAD - --single-
instance @@ %F @@
Icon=/home/christoph/.icons/FreeCAD_Logo.svg
Terminal=false
Type=Application
Icon=org.freecad.FreeCAD
Categories=Development;Engineering;
StartupNotify=true
StartupWMClass=FreeCAD
MimeType=application/x-extension-
fcstd;model/obj;image/vnd.dwg;image/vnd.dxf;model/vnd.collada+xml;appli
cation/iges;model/iges;model/step;model/step+zip;model/stl;application/
vnd.shp;model/vrml;
X-Flatpak=org.freecad.FreeCAD
Hidden=false
NoDisplay=false
```

```
mv org.freecad.FreeCAD.desktop ~/.local/share/applications
wget https://upload.wikimedia.org/wikipedia/commons/1/1a/FreeCAD-symbol.svg
-o FreeCAD_Logo.svg
mv FreeCAD_Logo.svg ~/.icons/
```

```
#Window Class ermitteln:
xprop WM_CLASS
# Dann "klick" auf die Titelleiste des unbekannten Fensters

# Validate *.desktop files
desktop-file-validate ~/.local/share/applications/<MYFILE>.desktop
```

Systembasis

Quelle(n):

- https://wiki.magenbrot.net/books/distributionen/page/vorlage-fur-apt-sourceslist-aktuell-debian-

trixie

Liste der Paketquellen auf das neue Format umstellen:

```
apt modernize-sources
```

- TUXEDO Software-Paketquellen hinzufügen

```
wget -O - https://deb.tuxedocomputers.com/0x54840598.pub.asc | gpg --dearmor
| sudo tee /usr/share/keyrings/tuxedo-archive-keyring.gpg > /dev/null
wget https://deb.tuxedocomputers.com/0x54840598.pub.asc
mv 0x54840598.pub.asc tuxedo-archive-keyring.gpg
cp tuxedo-archive-keyring.gpg /etc/apt/trusted.gpg.d/
#
apt-get install tuxedo-control-center
apt-get install tuxedo-drivers
apt-get install tuxedo-tomte-light
```

```
# deb cdrom:[Debian GNU/Linux 13.0.0 _Trixie_ - Official amd64 NETINST with
firmware 20250809-11:20]/ trixie contrib main non-free-firmware

deb https://debian.tu-bs.de/debian/ trixie main contrib non-free non-free-
firmware
deb-src https://debian.tu-bs.de/debian/ trixie main contrib non-free non-
free-firmware

deb https://security.debian.org/debian-security/ trixie-security main
contrib non-free non-free-firmware
deb-src https://security.debian.org/debian-security/ trixie-security main
contrib non-free non-free-firmware

# trixie-updates, to get updates before a point release is made;
# see
https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_updates_an
d_backports
deb https://debian.tu-bs.de/debian/ trixie-updates main contrig non-free
non-free-firmware
deb-src https://debian.tu-bs.de/debian/ trixie-updates main contrib non-free
non-free-firmware

# This system was installed using removable media other than
# CD/DVD/BD (e.g. USB stick, SD card, ISO image file).
# The matching "deb cdrom" entries were disabled at the end
# of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.

deb https://deb.tuxedocomputers.com/debian trixie main
```

```
apt-get update && apt-get dist-upgrade
apt-get install mc synaptic aptitude cifs-utils ca-certificates davfs2
```

```
libfuse2t64 gocryptfs gparted cryptmount partitionmanager putty dosfstools
mtools fatcat fatresize curl gedit gnome-tweaks dconf-editor gnome-tweaks
nemo nemo-data nemo-fileroller nemo-compare nemo-python yaru-theme-icon
yaru-theme-gnome-shell yaru-theme-unity gnome-icon-theme gnome-brave-icon-
theme gnome-commander rsync gnome-firmware dos2unix network-manager-openvpn-
gnome openvpn-systemd-resolved network-manager-vpnc-gnome
```



Paket-Datenbank neu aufbauen:
`sudo rm /var/lib/apt/lists/*`
`sudo apt update`

Nützliche Werkzeuge

```
apt-get install openjdk-21-jdk kdiff3 krename qtqr mediainfo mediainfo-gui
texlive texlive-lang-german texlive-latex-extra texmaker sqlitebrowser
```

Caja

Caja ist die Nemo Alternative aus der Linux-Mint Welt 😉

```
su
apt-get install caja caja-actions caja-actions-common caja-common caja-
extensions-common caja-image-converter caja-mediainfo caja-open-terminal
caja-rename
```

User -> ''sudo''

```
su -
usermod -aG sudo <username>
echo "<username>   ALL=(ALL:ALL) ALL" >> /etc/sudoers
apt install sudo
```

GoCryptFS

Zum Einbinden mit GoCryptFs verschlüsselter Verzeichnisse muss der Parameter „user_allow_other“ in der Datei `/etc/fuse.conf` eingetragen sein:

```
su
echo "user_allow_other" >> /etc/fuse.conf
```

Konfiguration

```
# init and use it:
mkdir ~/.gocryptfs ~/Chr_decrypt
gocryptfs -init ~/.gocryptfs
```

Autostart@Login

```
cp ./local/share/applications/MyMounts.desktop ./config/autostart/
chmod +x ./config/autostart/MyMounts.desktop
# disable the following lines by adding '#':
#  #Actions=...
#  #[Desktop ...
#  #Name= ...
```

Flatpak installieren und aktivieren



Flatpak ist eine Sandbox-Umgebung, d.h jede darin installierte Applikation läuft für sich (gekappselt) und hat, ohne Änderungen der Konfiguration (siehe `Flatseal`), keinen Zugriff auf z.B. Systemdienste, Geräte, oder das Home-Verzeichnis.

```
su
apt-get update && apt-get dist-upgrade
apt-get install flatpak gnome-software-plugin-flatpak
flatpak remote-add --if-not-exists flathub
https://dl.flathub.org/repo/flathub.flatpakrepo
```

Flatpak Pakete suchen
Flatpak Doku



Installierte Flatpak Pakete anzeigen: `flatpak list`
Pakete entfernen: `flatpak uninstall org.mozilla.Thunderbird`

Flatseal

Individuelles Rechte-Management mit Flatseal:

```
flatpak install flathub com.github.tchx84.Flatseal
```



Flatseal legt individuelle Konfigurationen unter



~/.local/share/flatpak/overrides/<Paketname> ab.

WebDAV

```
su -
usermod -aG davfs2 <benutzername>
```

KeepassXC

```
flatpak install flathub org.keepassxc.KeePassXC
```

GNOME Shell Erweiterungen

Unterstützung für GNOME Shell Erweiterungen installieren:

```
apt-get install gnome-shell gnome-shell-common gnome-shell-extension-apps-
menu gnome-shell-extension-auto-move-windows gnome-shell-extensions gnome-
shell-extension-desktop-icons-ng gnome-shell-extension-drive-menu gnome-
shell-extension-gpaste gnome-shell-extension-launch-new-instance gnome-
shell-extension-light-style gnome-shell-extension-manager gnome-shell-
extension-native-window-placement gnome-shell-extension-places-menu gnome-
shell-extension-prefs gnome-shell-extension-screenshot-window-sizer gnome-
shell-extension-status-icons gnome-shell-extension-system-monitor gnome-
shell-extension-tiling-assistant gnome-shell-extension-user-theme gnome-
shell-extension-window-list gnome-shell-extension-windows-navigator gnome-
shell-extension-workspace-indicator gnome-shell-extensions gnome-shell-
extensions-common gnome-shell-extensions-extra yaru-theme-gnome-shell gnome-
browser-connector gnome-extra-icons
```

Via `Erweiterungs-Manager` (gnome-shell-extension-manager) folgende GNOME Shell Erweiterungen installieren:

- Apps Menu
- Bluetooth Battery Meter
- Dash to Dock (by michele_g)
- Desktop Icons NG (DING) (by rastersof)
- Edit Desktop Files (.desktop)
- GSConnect (by dlandau)
- Lock Keys (by kazimieras.vaina)
- Net speed Simplified (by Prateek SU)
- Open Bar (by x10shun)
- Places Status Indicator
- Quick Settings Audio Panel
- Removable Drive Menu/
- Simple net speed (by bijignom) inkompatibel, 08/2025
- System Monitor (by fmuellner)

- Top Bar Organizer (zum Verschieben der LAN IP Anzeige in den linken Bereich)
- User Themes
- Just Perfection (by JustPerfection)

Pimp my Desktop - Conky

- https://github.com/brndnmtthws/conky
- conky.cc

```
cat .conkyrc
-- Conky, a system monitor https://github.com/brndnmtthws/conky
--
-- This configuration file is Lua code. You can write code in here, and it
will
-- execute when Conky loads. You can use it to generate your own advanced
-- configurations.
--
-- Try this (remove the `--`):
--
--   print("Loading Conky config")
--
-- For more on Lua, see:
-- https://www.lua.org/pil/contents.html
--
-- Conky Lua API: https://conky.cc/lua

-- Configuration settings: https://conky.cc/config_settings
conky.config = {
    alignment = 'middle_right',
    background = false,
    border_width = 1,
    cpu_avg_samples = 2,
    default_bar_height = 6,
    default_color = 'white',
    default_outline_color = 'white',
    default_shade_color = 'white',
    double_buffer = true,
    draw_borders = false,
    draw_graph_borders = true,
    draw_outline = false,
    draw_shades = false,
    extra_newline = false,
    font = 'DejaVu Sans Mono:size=11',
    format_human_readable = true,
    gap_x = 20,
    gap_y = 0,
    minimum_height = 5,
    minimum_width = 400,
    net_avg_samples = 2,
    no_buffers = true,
```

```
    out_to_console = false,
    out_to_ncurses = false,
    out_to_stderr = false,
    out_to_wayland = false,
    out_to_x = true,
    own_window = true,
    own_window_class = 'Conky',
    own_window_type = 'normal',
    own_window_transparent = true,
    own_window_argb_visual = true,
    own_window_argb_value = 145,
    own_window_hints = 'undecorated,sticky,below,skip_taskbar,skip_pager',
    show_graph_range = false,
    show_graph_scale = true,
    short_units = true,
    stippled_borders = 0,
    update_interval = 1.0,
    uppercase = false,
    use_spacer = 'none',
    use_xft = true,

    --Colours
--    default_color = '#D9FFE2',
      color1 = '#FF0000',     --red
      color2 = '#3E5570',     --grey/blue
      color3 = '#CCCCCC',     --grey
      color4 = '#BBBBBB',     --grey
      color5 = '#C0FF00',     --light green
      color6 = '#FFFFFF',     --white

    --Signal Colours
      color7 = '#C0FF00',         --light green
      color8 = '#FFA726',         --orange
      color9 = '#F1544B',         --firebrick

}

-- Variables: https://conky.cc/variables
conky.text = [[
${color6}${font Roboto:size=30}${alignc}${time %H}:${time %M}${font}
${color6}${font Roboto:size=15}${alignc}${time %A}, ${time %e} ${time %B}
${time %Y}${font}

${color6}${font Roboto:style=Bold:size=15}S Y S T E M ${hr 2}${font}
${color grey}OS: ${distribution} $sysname
${color grey}Kernel: ${kernel} ${sysname}
${color grey}Uptime:$color $uptime

${color6}${font Roboto:style=Bold:size=15}C P U ${hr 2}${font}
${color grey}Frequency (in GHz):$color $freq_g
```

```
${color grey}CPU  1 usage:$color ${cpu cpu1} ${cpubar cpu1}
${color grey}CPU  2 usage:$color ${cpu cpu2} ${cpubar cpu2}
${color grey}CPU  3 usage:$color ${cpu cpu3} ${cpubar cpu3}
${color grey}CPU  4 usage:$color ${cpu cpu4} ${cpubar cpu4}
${color grey}CPU  5 usage:$color ${cpu cpu5} ${cpubar cpu5}
${color grey}CPU  6 usage:$color ${cpu cpu6} ${cpubar cpu6}
${color grey}CPU  7 usage:$color ${cpu cpu7} ${cpubar cpu7}
${color grey}CPU  8 usage:$color ${cpu cpu8} ${cpubar cpu8}
${color grey}CPU  9 usage:$color ${cpu cpu9} ${cpubar cpu9}
${color grey}CPU 10 usage:$color ${cpu cpu10} ${cpubar cpu10}
${color grey}CPU 11 usage:$color ${cpu cpu11} ${cpubar cpu11}
${color grey}CPU 12 usage:$color ${cpu cpu12} ${cpubar cpu12}

${color6}${font Roboto:style=Bold:size=15}M E M O R Y ${hr 2}${font}
${color grey}RAM Usage:$color $mem / $memmax - $memperc% ${membar 4}
${color grey}Swap Usage:$color $swap / $swapmax - $swapperc% ${swapbar 4}

${color6}${font Roboto:style=Bold:size=15}F I L E  S Y S T E M ${hr
2}${font}
 /: $color${fs_used /} / ${fs_size /} ${fs_bar 6 /}
 /home: $color${fs_used /home} / ${fs_size /home} ${fs_bar 6 /home}

${color6}${font Roboto:style=Bold:size=15}B A T T E R Y ${hr 2}${font}
${color grey}Power Consumption: ${battery_power_draw} W
${color grey}State: ${battery_status}, Runtime: ${battery_time}
${color grey}Capacity: ${battery_percent}% ${battery_bar}

${color6}${font Roboto:style=Bold:size=15}P R O C E S S E S ${hr 2}${font}
${color grey}Processes:$color $processes  ${color grey}Running:$color
$running_processes
$hr
${color grey}Name                 PID    CPU%   MEM%
${color lightgrey} ${top name 1} ${top pid 1} ${top cpu 1} ${top mem 1}
${color lightgrey} ${top name 2} ${top pid 2} ${top cpu 2} ${top mem 2}
${color lightgrey} ${top name 3} ${top pid 3} ${top cpu 3} ${top mem 3}
${color lightgrey} ${top name 4} ${top pid 4} ${top cpu 4} ${top mem 4}

${if_up wlo1}
${color6}${font Roboto:style=Bold:size=15}N E T W O R K  ${hr 2}${font}
${color6}Internal IP Address:${color}${alignr}${addrs wlo1}
${color6}Default Gateway:${color}${alignr}${gw_ip}
${color6}External IP Address:${color}${alignr}${texeci 3600 wget -q -O-
https://ipecho.net/plain}
${color6}WLAN SSID: ${alignr}${wireless_essid wlo1}
${color6}WLAN Frequency: ${alignr}${wireless_freq wlo1}
${color6}WLAN Bitrate: ${alignr}${wireless_bitrate wlo1}
${color6}WLAN Link Quality: ${alignr}${wireless_link_qual_perc wlo1}%
${color6}Network Speed (Up/Down):${color}${alignr}${upspeed wlo1} /
${downspeed wlo1}
${color6}Upload / Download Total:${color}${alignr}${totalup wlo1} /
${totaldown wlo1}
```

```
${upspeedgraph wlo1 35, 400 ff0000 00ff00 100000000 -t -l}
${downspeedgraph wlo1 35, 400 ff0000 00ff00 100000000 -t -l}
${endif}

${if_up enx00e04c6808e4}
${color6}${font Roboto:style=Bold:size=15}N E T W O R K  ${hr 2}${font}
${color6}Internal IP Address:${color}${alignr}${addrs enx00e04c6808e4}
${color6}Default Gateway:${color}${alignr}${gw_ip}
${color6}External IP Address:${color}${alignr}${texeci 3600 wget -q -O-
https://ipecho.net/plain}
${color6}Network Speed (Up/Down):${color}${alignr}${upspeed enx00e04c6808e4}
/ ${downspeed enx00e04c6808e4}
${color6}Upload / Download Total:${color}${alignr}${totalup enx00e04c6808e4}
/ ${totaldown enx00e04c6808e4}
${upspeedgraph enx00e04c6808e4 35, 400 ff0000 00ff00 100000000 -t -l}
${downspeedgraph enx00e04c6808e4 35, 400 ff0000 00ff00 100000000 -t -l}
${endif}
]]
```

Yubico Authenticator

```
flatpak install flathub com.yubico.yubioath
```

Nitrokey

```
sudo add-apt-repository ppa:nitrokey/nitrokey
sudo apt install nitrokey-app
#
flatpak install flathub com.nitrokey.nitrokey-app2
```

AusweisApp2



Konfig-Datei liegt unter: `~/.var/app/de.bund.ausweisapp.ausweisapp2/config/flathub/AusweisApp2.conf`

```
flatpak install flathub de.bund.ausweisapp.ausweisapp2
```

KDE ISO Imager

Warpinator

```
flatpak install flathub org.x.Warpinator
```

FileZilla



Konfig-Datei liegt unter:
~/.var/app/org.filezillaproject.Filezilla/config/filezilla/
Die Flatpak Konfig liegt hier:
~/.local/share/flatpak/overrides/org.filezillaproject.Filezilla

```
flatpak install flathub org.filezillaproject.Filezilla
```

org.filezillaproject.Filezilla

```
[Context]
sockets=!ssh-auth
filesystems=home
```

Docker

Quelle(n):

- https://docs.docker.com/engine/install/debian/

```
sudo su
sudo apt remove $(dpkg --get-selections docker.io docker-compose docker-doc
podman-docker containerd runc | cut -f1)
apt install ca-certificates curl
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
#
sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: https://download.docker.com/linux/debian
Suites: $(. /etc/os-release && echo "$VERSION_CODENAME")
Components: stable
Signed-By: /etc/apt/keyrings/docker.asc
EOF
#
apt update
#
apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
#
sudo systemctl status docker
```

```
#
sudo systemctl start docker
```

Portainer CE

Quelle(n):

- https://docs.portainer.io/start/install-ce/server/docker/linux

```
###
```

Office und Internet

Brother HL-3170CDW Farblaserdrucker

Quelle: https://www.brother.de/support/hl-3170cdw/downloads

```
cd Downloads
wget https://download.brother.com/welcome/dlf006893/linux-brprinter-
installer-2.2.6-0.gz
gunzip linux-brprinter-installer-2.2.6-0.gz
chmod +x linux-brprinter-installer-2.2.6-0
sudo su
./linux-brprinter-installer-2.2.6-0 HL-3170CDW
```

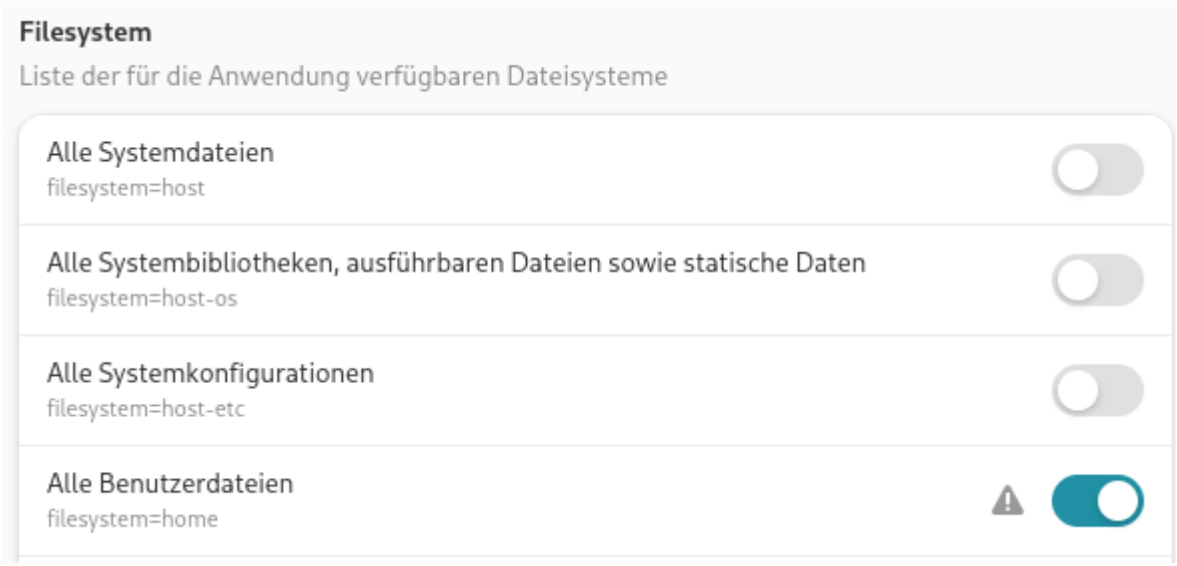
Drucker über CUPS hinzufügen und konfigurieren:
http://localhost:631/

- **Verwaltung** -> **Drucker hinzufügen** -> LPD/LPR-Host oder -Drucker -> **Weiter**
- **Verbindung:** -> lpd://192.168.100.8/BINARY_P1 -> **Weiter**
- **Name:** -> Brother_HL3170CDW, **Beschreibung** -> Farblaserdrucker, **Ort** -> Büro -> **Weiter**
- **Hersteller** -> BROTHER -> **Weiter**
- **Modell:** -> Brother HL-3170CDW -> **Drucker hinzufügen**

Firefox

```
flatpak install flathub org.mozilla.firefox
```

Anschließend via Flatseal das Home-Verzeichnis zugänglich machen:



Oder alternativ:

```
cd ~
echo -e '[Context]\ndevices=dri\nfilesystems=home' >>
.local/share/flatpak/overrides/org.mozilla.firefox
```

Benutzerdefinierte Tabs: `~./mozilla/firefox/<PROFILE-FOLDER>/chrome/userChrome.css`

userChrome.css

```
/* Optional: Zusätzlicher Abstand zwischen den Tabs */
.tabbrowser-tab + .tabbrowser-tab {
    margin-top: -6px !important; /* oder ein anderer gewünschter
Abstand */
}

#TabsToolbar {
  --tab-min-height: 20px; /*  Passen Sie diesen Wert an (z.B. 40px,
50px) */
  --tab-max-height: 20px;
}

#tabbrowser-tabs {
  --tab-min-height: 20px; /*  Gleicher Wert wie oben */
  --tab-max-height: 20px;
}
```

Video Downloadhelper

```
# Companion App herunterladen:
# https://github.com/aclap-dev/video-downloadhelper/wiki/CoApp-Installation
wget https://github.com/aclap-
dev/vdhcoapp/releases/latest/download/vdhcoapp-linux-x86_64.deb
sudo dpkg -i vdhcoapp-linux-x86_64.deb
```

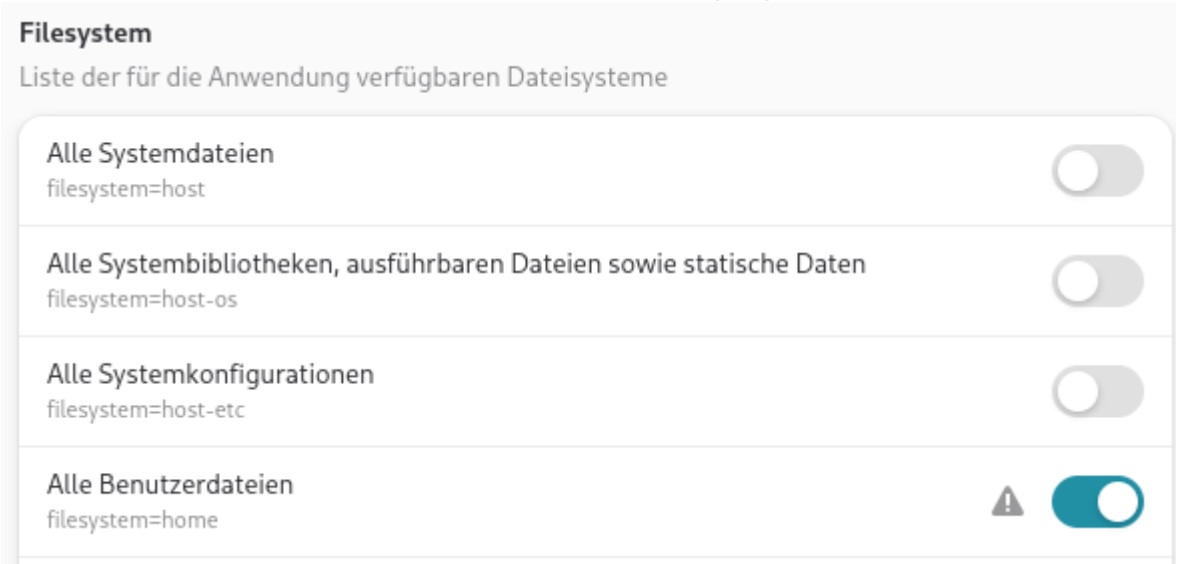
```
/opt/vdhcoapp/vdhcoapp install # (not as root! Don't use sudo)
Installing…
Flatpak is installed. Making the coapp available from browser sandboxes:
Linked coapp within org.mozilla.firefox.
Linked coapp within com.brave.Browser.
Linked coapp within com.google.Chrome.
Linked coapp within com.google.ChromeDev.
Linked coapp within org.chromium.Chromium.
Linked coapp within com.github.Eloston.UngoogledChromium.
Linked coapp within com.microsoft.Edge.
Linked coapp within com.microsoft.EdgeDev.
Writing /home/christoph/.mozilla/native-messaging-
hosts/net.downloadhelper.coapp.json
Writing /home/christoph/.var/app/org.mozilla.firefox/.mozilla/native-
messaging-hosts/net.downloadhelper.coapp.json
Writing /home/christoph/.config/google-
chrome/NativeMessagingHosts/net.downloadhelper.coapp.json
Writing
/home/christoph/.config/chromium/NativeMessagingHosts/net.downloadhelper.coa
pp.json
Writing /home/christoph/.config/BraveSoftware/Brave-
Browser/NativeMessagingHosts/net.downloadhelper.coapp.json
Writing /home/christoph/.var/app/com.google.Chrome/config/google-
chrome/NativeMessagingHosts/net.downloadhelper.coapp.json
VdhCoApp : VdhCoApp is ready to be used
```

Thunderbird

Aufbau der ''profiles.ini''

```
flatpak install flathub org.mozilla.Thunderbird
```

Anschließend via `Flatseal` das Home-Verzeichnis zugänglich machen:



Oder alternativ:

```
cd ~
echo -e '[Context]\nfilesystems=home' >>
.local/share/flatpak/overrides/org.mozilla.Thunderbird
```

Nach der Installation einfach den `.thunderbird` Ordner aus dem Backup ins Home-Verzeichnis zurück kopieren.

Hibiscus



Konfig-Datei liegt unter: `~/.jameica.properties`
Datensatz liegt unter `~/.jameica`

Download direkt vom Hersteller: https://www.willuhn.de/products/jameica/download.php
Entpacken, `jameica.sh` starten ...

```
wget
https://www.willuhn.de/products/jameica/releases/current/jameica/jameica-
linux64-2.10.5.zip
unzip jameica-linux64-2.10.5.zip
chmod +x jameica.sh
./jameica.sh
```

Plugin `Hibiscus` installieren.

Threema

```
flatpak install flathub ch.threema.threema-web-desktop
```

Signal

```
flatpak install flathub org.signal.Signal
```

Telegram

Joplin

LaTex

Adobe Reader

QOwnNotes

Nextcloud Client

Converseen

LibreOffice

```
flatpak install flathub org.libreoffice.LibreOffice
```

qpdfview



qpdfview habe ich durch **Okular** ersetzt

Angepasste Einstellungen:
Datei-Werkzeugleiste: openInNewTab,refresh,saveAn,print
Bearbeiten-Werkzeugleiste: firstPage,currentPage,previousPage,nextPage,search,copyToClipboardMode,addAnnotationMode
Ansicht-Werkzeugleiste: scaleFactor,zoomIn,zoomOut, continuousMode,twoPagesModem,originalSize,fitToPageWidthMode,fitToPageSizeMode,rotateLeft,rotateRight,fullscreen

Okular

```
flatpak install flathub org.kde.okular
```

Pinta

```
flatpak install flathub com.github.PintaProject.Pinta
```

XnView MP

```
flatpak install flathub com.xnview.XnViewMP
```

Entwicklung

FreeCAD

```
flatpak install flathub org.freecad.FreeCAD
```

Prusa Slicer



Konfig-Pfad: ~/.var/app/com.prusa3d.PrusaSlicer/config/PrusaSlicer

```
flatpak install flathub com.prusa3d.PrusaSlicer
```

Visual Studio Code

```
flatpak install flathub com.visualstudio.code
```

Arduino 2.x

MQTT Explorer

```
flatpak install flathub io.github.Omniaevo.mqtt5-explorer
```

VirtualBox

Quelle: https://www.virtualbox.org/wiki/Downloads

```
wget
https://download.virtualbox.org/virtualbox/7.2.0/virtualbox-7.2_7.2.0-170228
~Debian~trixie_amd64.deb
wget
https://download.virtualbox.org/virtualbox/7.2.0/Oracle_VirtualBox_Extension
_Pack-7.2.0.vbox-extpack
sudo su
apt-get install libqt6help6 gcc make perl linux-headers-amd64 linux-
headers-6.12.41+deb13-amd64
dpkg -i virtualbox-7.2_7.2.0-170228~Debian~trixie_amd64.deb
```

Wine32/64

```
sudo
dpkg --add-architecture i386
apt update
```

```
apt install wine wine32 libwine:i386 fonts-wine
```

GUIscrcpy

Um der `scrcpy` oder per `GUIscrcpy` auf ein Android Smartphone zugreifen zu können, muss bei diesem zuvor der Debug-Modus aktiviert werden. Dazu müssen die sog. Entwickleroptionen freigeschaltet werden. Soll der Zugriff nur über USB erfolgen, reicht die Freigabe von „Debugging über USB“ aus. Es geht drahtlos etwas komfortabler über WLAN mit „Debugging über WLAN“.

```
sudo su
apt-get install adb
exit
flatpak install flathub in.srev.guiscrcpy
```

Multimedia

VLC



Konfig-Datei: `~/.var/app/org.videolan.VLC/config/vlc/`

```
flatpak install flathub org.videolan.VLC
```

MediathekViewer



Konfig liegt unter: `~/.var/app/de.mediathekview.MediathekView/config/mediathek3/`

```
flatpak install flathub de.mediathekview.MediathekView
```

gPodder

AviDemux

Asunder

RipperX

Audacity

EasyTag

MediaInfo

Gaming

Steam

From:
https://www.von-thuelen.de/ - **Christophs DokuWiki**

Permanent link:
https://www.von-thuelen.de/doku.php/wiki/linux/debian_13_trixie/uebersicht

Last update: **2026/03/25 16:38**