

Software

Tools to aid you

- [Spell checking](#)
- [Fonts](#)
- [Polkit](#)
- [Firefox](#)
- [Google Chrome](#)
- [Discord](#)
- [Blu-ray](#)
- [Node.js \(nvm\)](#)
- [KVM](#)
- [Folding@Home](#)
- [Timeshift](#)
- [GNOME Flatpaks](#)

Spell checking

Hunspell is a spell checker and morphological analyzer library used by Firefox, Thunderbird, Chromium, LibreOffice and more.

Install the following packages to enable system-wide spell checking and hyphenation support (add languages for `hunspell` and `hyphen` at your discretion):

```
pacman -S hunspell hunspell-de hunspell-en_US hyphen hyphen-de hyphen-en
```

Fonts

For most desktop environments, a sufficient number of fonts is installed as dependencies. However, there's several additional packages for different styles and writing systems (latin vs. non-latin scripts). [Arch Wiki](#) has an extensive list of available fonts in both the repositories and the AUR. Installing the Noto font family also provides a vast coverage over a large array of scripts.

Configuration

Most applications read the font configuration provided by the `fontconfig` library. These configurations are written in XML and read from several different locations.

Location	Description
<code>/etc/fonts/fonts.conf</code>	Master configuration file (not for editing!)
<code>/etc/fonts/conf.d</code>	System-wide additional drop-in configuration files, hand-written or as symbolic links
<code>\$XDG_CONFIG_HOME/fontconfig/fonts.conf</code>	Per-user config file
<code>\$XDG_CONFIG_HOME/fontconfig/conf.d</code>	Per-user additional drop-in configuration files, hand-written or as symbolic links

Configuration files are read in and applied in lexical order. If you need rules applied in a specific order, make sure to prepend them with 2-digit numbers in the order you need.

A minimal `fontconfig` configuration file contains these headers:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "urn:fontconfig:fonts.dtd">
<fontconfig>

  <!-- settings go here -->

</fontconfig>
```

Some font packages come with pre-defined rule sets, which are installed to `/usr/share/fontconfig/conf.avail/`. To apply them, it's best to create symbolic links to them in their respective drop-in configuration directories.

To apply them system-wide, link them from the `/etc/fonts/conf.d` directory:

```
cd /etc/fonts/conf.d
sudo ln -s /usr/share/fontconfig/conf.avail/70-no-bitmaps-except-emoji.conf
```

To apply them only to the currently logged in user, link them in the

`$XDG_CONFIG_HOME/fontconfig/conf.d` directory:

HINT: The environment variable `$XDG_CONFIG_HOME` should point to the `.config` sub-directory in your home directory. If it doesn't, use `$HOME/.config` instead for the examples or set it with `export`.

```
mkdir $XDG_CONFIG_HOME/fontconfig/conf.d
ln -s /usr/share/fontconfig/conf.avail/70-no-bitmaps-except-emoji.conf
$XDG_CONFIG_HOME/fontconfig/conf.d
```

Emoji Fonts

There are a few emoji fonts available on Arch.

Name	Package	Description
JoyPixels	<code>ttf-joypixels</code>	formerly EmojiOne, part of Emoji as a Service, proprietary
Noto Color Emoji	<code>noto-fonts-emoji</code>	Google open-source emoji font, color
Twemoji (Twitter Emoji)	<code>ttf-twemoji</code> (AUR)	Emoji for everyone, originally created by Twitter

Install your selected emoji font:

```
pacman -S noto-fonts-emoji
```

Applications requesting emoji to be displayed should pick up on the font after restarting them.

NOTE: KDE sometimes applies emoji fonts incorrectly, either not showing them at all or showing the outline symbol version from a different font. You can fix this by installing `noto-color-emoji-fontconfig` from the AUR and creating a symbolic link to the configuration file as shown above.

Polkit

`polkit` is an application-level toolkit for defining and handling the policy that allows unprivileged processes to speak to privileged processes: It is a framework for centralizing the decision making process with respect to granting access to privileged operations for unprivileged applications.

Custom rules

Mount disks as user

Edit/create `/etc/polkit-1/rules.d/50-udisk.rules`

```
// Original rules: https://github.com/coldfix/udiskie/wiki/Permissions
// Changes: Added org.freedesktop.udisks2.filesystem-mount-system, as this is used by Dolphin.

polkit.addRule(function(action, subject) {
  var YES = polkit.Result.YES;
  // NOTE: there must be a comma at the end of each line except for the last:
  var permission = {
    // required for udisksl:
    "org.freedesktop.udisks.filesystem-mount": YES,
    "org.freedesktop.udisks.luks-unlock": YES,
    "org.freedesktop.udisks.drive-eject": YES,
    "org.freedesktop.udisks.drive-detach": YES,
    // required for udisk2:
    "org.freedesktop.udisks2.filesystem-mount": YES,
    "org.freedesktop.udisks2.encrypted-unlock": YES,
    "org.freedesktop.udisks2.eject-media": YES,
    "org.freedesktop.udisks2.power-off-drive": YES,
    // Dolphin specific
    "org.freedesktop.udisks2.filesystem-mount-system": YES,
    // required for udisk2 if using udiskie from another seat (e.g. systemd):
    "org.freedesktop.udisks2.filesystem-mount-other-seat": YES,
    "org.freedesktop.udisks2.filesystem-unmount-others": YES,
    "org.freedesktop.udisks2.encrypted-unlock-other-seat": YES,
    "org.freedesktop.udisks2.eject-media-other-seat": YES,
    "org.freedesktop.udisks2.power-off-drive-other-seat": YES
  }
});
```

```
};  
if (subject.isInGroup("storage")) {  
    return permission[action.id];  
}  
});
```

Firefox

Install Firefox via these packages (adjust for your desired locale):

```
pacman -S firefox firefox-i18n-de
```

Wayland

Wayland is not yet the default display manager for Firefox (it falls back to XWayland on Wayland). To force Firefox to use Wayland you can set the `MOZ_ENABLE_WAYLAND` environment variable to `1`. Use user specific systemd environment variable configs to set it:

```
echo "MOZ_ENABLE_WAYLAND=1" >> ~/.config/environment.d/moz_wayland.conf
```

Media Playback

Autoplay in background

Firefox prevents autoplay for media of tabs that aren't currently active, which causes apps like Plex to take very long to skip to the next track after the current one has ended. The following setting in `about:config` can be used to disable this behavior:

Setting key	Value	Description
<code>media.block-autoplay-until-in-foreground</code>	<code>false</code>	Enable autoplay when tab is not currently active

Hardware Decoding

Utilizing GPU hardware accelerated decoding of video content results in smoother playback of HD/4K content, while reducing CPU load and power draw (important to save on battery on laptops).

To ensure Firefox uses hardware decoding ensure the following:

- The necessary VA-API drivers are installed (see: [Graphics Cards](#))
- Navigate to `about:support` and ensure that under *Compositing* it says *WebRender* (*WebRender Software* will **not** work)

- Navigate to `about:config` and set `media.ffmpeg.vaapi.enabled` to `true`
- If running Wayland, enable Wayland mode in Firefox (see above)

Verify hardware video decoding

To verify Firefox is actually using VA-API to decode video you can launch it with the following command:

```
MOZ_LOG="FFmpegVideo:5" firefox 2>&1 | grep 'VA-API'
```

Start playing some video in Firefox and watch the logs on your terminal. If your log output reads something like the following video decoding via VA-API is working.

```
[RDD 97685: MediaPDecoder #1]: D/FFmpegVideo FFVPX: Initialising VA-API FFmpeg decoder
[RDD 97685: MediaPDecoder #2]: D/FFmpegVideo FFVPX:  VA-API FFmpeg init successful
[RDD 97685: MediaPDecoder #2]: D/FFmpegVideo FFVPX: Choosing FFmpeg pixel format for VA-API
video decoding.
[RDD 97685: MediaPDecoder #1]: D/FFmpegVideo FFVPX:  VA-API FFmpeg init successful
[RDD 97685: MediaPDecoder #2]: D/FFmpegVideo FFVPX: VA-API Got one frame output with pts=0
dts=0 duration=40000 opaque=-9223372036854775808
[RDD 97685: MediaPDecoder #1]: D/FFmpegVideo FFVPX: Initialising VA-API FFmpeg decoder
[RDD 97685: MediaPDecoder #1]: D/FFmpegVideo FFVPX:  VA-API FFmpeg init successful
[RDD 97685: MediaPDecoder #1]: D/FFmpegVideo FFVPX: VA-API Got one frame output with pts=40000
dts=40000 duration=40000 opaque=-9223372036854775808
[RDD 97685: MediaPDecoder #2]: D/FFmpegVideo FFVPX: VA-API Got one frame output with pts=80000
dts=80000 duration=40000 opaque=-9223372036854775808
[RDD 97685: MediaPDecoder #2]: D/FFmpegVideo FFVPX: VA-API Got one frame output with
pts=120000 dts=120000 duration=40000 opaque=-9223372036854775808
```

KDE Plasma Integration

For better integration of Firefox into the KDE Plasma desktop, install the Plasma Integration add-on either via the [Mozilla Add-on page](#). It enables rich notifications support and download progress integration into the notification area of KDE Plasma.

Media Playback Controls

To prevent duplicate entries in the Media Player widget or tray icon, set `media.hardwaremediakeys.enabled` to `false`. This disables the media entry from Firefox itself and only uses the one from the Plasma integration add-on.

KDE Dialogs

By default, Firefox uses GTK file and print dialogs, even on KDE. To change this to KDE native dialogs navigate to `about:config` and change the appropriate `widget.use-xdg-desktop-portal` settings to `1`.

Google Chrome

Install Google Chrome from AUR:

```
yay -S google-chrome
```

Tweaks

To enable hardware accelerated video decoding (with open source drivers) create a file at `~/.config/chrome-flags.conf` and add the following line in it:

```
--enable-features=VaapiVideoDecoder
```

Additionally, if you need to be able to share your screen wie WebRTC, you need to add the following line as well:

```
--enable-usermedia-screen-capturing
```

Furthermore, visit <chrome://flags> and set the following options to further tweak performance (use the search field to filter):

Setting key	Value	Description
<code>#enable-webrtc-pipewire-capturer</code>	Enabled	Uses PipeWire to capture the screen in Wayland sessions
<code>#enable-gpu-rasterization</code>	Enabled	Uses GPU for rasterization, boosting performance
<code>#enable-zero-copy</code>	Enabled	Accesses GPU memory directly, boosting performance
<code>#ozone-platform-hint</code>	Auto	Auto-detects which windowing system is currently in use (X11, Wayland)

Discord

Discord is a proprietary, cross-platform, all-in-one voice and text chat application.

Install Discord from the repositories:

```
pacman -S discord
```

Screensharing on Wayland

Currently, it's not possible to reliably share your screen during a call with Discord on Wayland. The current beta builds have support for native screensharing, but sharing audio only works with `pulseaudio` as the primary sound server.

An alternative client, called Vesktop, has full native support for screen sharing with Discord on Wayland with audio. It can be used instead of the official Discord app.

Install it from the AUR:

```
yay -S vesktop
```

Blu-ray

Playback

In order play Blu-Rays install the following packages:

```
sudo pacman -S libbluray libaacs
```

Additionally, a `KEYDB.cfg` file is needed. Download it from the [FindVUK Online Database](#)

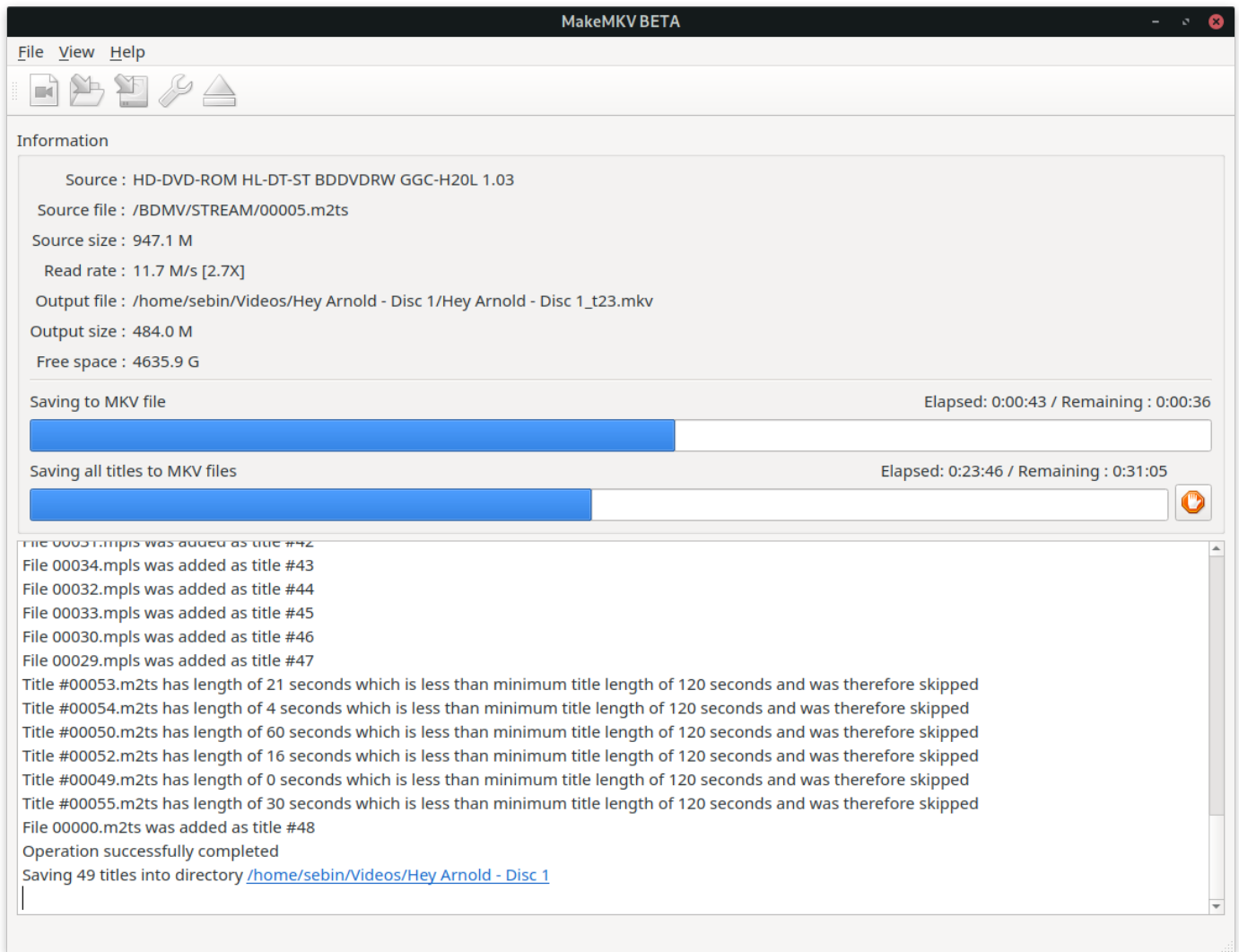
Extract the ZIP to `~/.config/aacs/`:

ATTENTION: You may need to rename the `keydb.cfg` file to `KEYDB.cfg` (lower to upper case) for tooling to find it.

```
unzip keydb_eng.zip -d ~/.config/aacs/
```

After that use any Blu-Ray capable playback software, e.g. `vlc bluray:///dev/sr0` to play back Blu-Rays.

Ripping



In order to rip Blu-Rays install [MakeMKV](#) from the AUR:

```
yay -S makemkv
```

MakeMKV requires the `sg` (*SCSI generic (sg) driver*) kernel module to be loaded in order to recognize the drive. To load the module temporarily:

```
sudo modprobe sg
```

To have the kernel load the module on each boot:

```
sudo echo sg > /etc/modules-load.d/sg.conf
```

Node.js (nvm)

Use the Node Version Manager (`nvm`) to install Node.js into your current user's path and switch Node.js versions on the fly.

Install `nvm` via the AUR:

```
yay -S nvm
```

Include the init script `/usr/share/nvm/init-nvm.sh` into your shell configuration to load it each time you start your terminal:

```
# bash
echo 'source /usr/share/nvm/init-nvm.sh' >> ~/.bashrc

# zsh
echo 'source /usr/share/nvm/init-nvm.sh' >> ~/.zshrc
```

Restart your terminal to reload all init scripts and you should be able to use `nvm` to install a Node.js version of your choice:

```
nvm install 12
```

Migrating globally installed `npm` packages

When you install and switch to a different `nvm` managed version of Node.js (`nvm install 14` or `nvm use 16`) you may find that your globally installed `npm` packages (e.g. `svggo`) are no longer available until you switch back to the specific version of Node.js you have been using before the upgrade or switch.

This is because globally installed `npm` packages are installed for the specific version of Node.js you happen to be using at the time of installation and placed in a directory i.e.

`~/.nvm/versions/node/v16.14.0/lib/node_modules`. When you install a different version, e.g. `17.2.0` the path to your Node.js installation changes to `~/.nvm/versions/node/v17.2.0/lib/node_modules`.

Use the `--reinstall-packages-from=<version>` option to carry over globally installed packages to the new Node.js installation.

You can either pass a specific version you want to reinstall globally installed packages from or use bash string expansion to reinstall from the currently active one in use:

```
nvm install <new version> --reinstall-packages-from=<old version>
```

```
nvm install 17 --reinstall-packages-from=$(node -v)
```

KVM

KVM (Kernel-based Virtual Machine) is a hypervisor built into the Linux kernel.

1. Install `libvirt` Packages

```
yay -S qemu libvirt edk2-ovmf virt-manager nfs-utils virtio-win

# optional dependencies
iptables-nft dnsmasq    # for default NAT/DHCP networking
bridge-utils           # for bridged networking
openbsd-netcat          # for remote management over SSH
```

2. Add user to `libvirt` groups

```
sudo usermod -aG libvirt $USER
```

3. Start `libvirtd` daemon

```
sudo systemctl enable --now libvirtd
```

4. Create network bridge with `nmcli`

```
nmcli connection add type bridge ifname br0 con-name "Netzwerkbrücke" stp no
nmcli connection add type bridge-slave ifname enp39s0 con-name "Ethernet" master br0
nmcli connection down "Kabelgebundene Verbindung 1"
nmcli connection up "Netzwerkbrücke"
```

- When using bonding of interfaces, disable IPv4 and IPv6 on the **bridge**

```
nmcli con mod "Netzwerkbrücke" ipv4.method disabled ipv6.method ignore
```

5. Define bridge network XML file, e.g. as `br0.xml`

```
<network>
  <name>br0</name>
  <forward mode='bridge'/>
  <bridge name='br0'/>
</network>
```

6. Add bridge network to `virt-manager`


```
virsh -c qemu:///system net-define br0.xml
virsh -c qemu:///system net-autostart br0
```

7. Disable COW on Btrfs (optional, recommended)

```
sudo chattr +C /var/lib/libvirt/images
```

8. Define a remote storage pool (e.g. remote ISO images) `remote-iso.xml`

```
<pool type="netfs">
  <name>iso</name>
  <source>
    <host name="dragonhoard"/>
    <dir path="/Download/Software/ISOs"/>
    <format type="auto"/>
  </source>
  <target>
    <path>/var/lib/libvirt/images/iso</path>
  </target>
</pool>
```

9. Add storage pool to `virt-manager`

```
virsh -c qemu:///system pool-define remote-iso.xml
virsh -c qemu:///system pool-autostart iso
```

10. Create the storage pool mountpoint

```
sudo mkdir -p /var/lib/libvirt/images/iso
```

Folding@Home

Help scientists studying Alzheimer's, Huntington's, Parkinson's, and SARS-CoV-2 by simply running a piece of software on your computer. Add your computer to a network of millions of others around the world to form the world's largest distributed supercomputer.

Installation

```
yay -S foldingathome opencl-amd
```

Configuration

Run `FAHClient --configure` as `root` to generate a configuration file at `/etc/foldingathome/config.xml`:

```
cd /etc/foldingathome
FAHClient --configure
```

Then start/enable the `foldingathome.service` systemd unit. NVIDIA users should also enable the `foldingathome-nvidia.service` systemd unit.

Example Configuration

```
<config>
  <!-- Slot Control -->
  <power v='FULL' />

  <!-- User Information -->
  <passkey v='1234567890' />
  <team v='45032' />
  <user v='Registered_User_Name' />

  <!-- Folding Slots -->
  <slot id='0' type='CPU' />
  <slot id='1' type='GPU' />
</config>
```


Timeshift

IMPORTANT: Timeshift is **not a backup tool!** It only creates *local snapshots* of the system to roll back changes to the system. Do not rely on this mechanism to keep your data safe! Timeshift deletes the oldest snapshot when a new one is created and the maximum number of snapshots is reached. Furthermore, if the underlying file system is corrupted, the snapshots will be, too! Use a proper backup tool to keep your data safe on external data storage!

Timeshift helps create incremental snapshots of the file system at regular intervals, which can then be restored at a later date to undo all changes to the system.

It supports `rsync` snapshots for all filesystems, and uses the built-in snapshot features for Btrfs drives configured to use the `@` and `@home` subvolume layout for *root* and *home* directories respectively.

Installation

Timeshift is available from the Arch repos. It uses cron to make regularly scheduled backups. Install Timeshift with a cron daemon, e.g. `cronie`:

```
pacman -S timeshift cronie
```

Start and enable the cron scheduler for Timeshift to take regular snapshots:

```
sudo systemctl enable --now cronie
```

Finally, start Timeshift and complete the first time setup.

Automatic snapshots on system changes

In addition to Timeshift's periodic snapshots, `timeshift-autosnap` provides a `pacman` hook to create a manual snapshot every time packages are installed, upgraded or removed.

Install `timeshift-autosnap` from the AUR:

```
yay -S timeshift-autosnap
```

By default `timeshift-autosnap` only keeps 3 snapshots. To change this, edit `/etc/timeshift-autosnap.conf` and either set `deleteSnapshots` to `false` to never delete any snapshots or increase the number of `maxSnapshots`:

```
skipAutosnap=false
deleteSnapshots=true
maxSnapshots=7
updateGrub=true
snapshotDescription={timeshift-autosnap} {created before upgrade}
```

Prevent excessive snapshotting when using `yay`

By default, when installing or updating multiple packages from the AUR, `yay` first builds a package and immediately calls `pacman` to install it, before building and installing the next one on its list. This also means that the `timeshift-autosnap` hook is triggered **for each individual AUR package** built by `yay`, **including dependencies also installed from the AUR**.

This can have undesirable side-effects:

- `yay` will cause `timeshift-autosnap` to reach the `maxSnapshots` limit very quickly when installing multiple packages from the AUR, leaving you with snapshots with little to no meaningful changes between them
- if `deleteSnapshots` is set to `false` the amount of snapshots might quickly exhaust the usable space on the drive

To prevent this it is recommended to configure `yay` to:

1. not remove make dependencies after successfully built packages are installed
2. build all AUR packages first, install them all later
3. install AUR packages together with regular repo packages

By calling `yay` with the `--save` parameter, any options passed to it will be saved in a configuration file, e.g.:

```
yay --noremovemake --batchinstall --combinedupgrade --save
```

Next time you use `yay` to install, upgrade or remove packages it will read the generated config file at `~/.config/yay/config.json` and apply the options automatically without having to specify them during use.

GNOME Flatpaks

Core apps

Name	ID	Description
Calculator	<code>org.gnome.Calculator</code>	Perform arithmetic, scientific or financial calculations
Calendar	<code>org.gnome.Calendar</code>	Manage your schedule
Calls	<code>org.gnome.Calls</code>	Make phone and SIP calls
Camera	<code>org.gnome.Snapshot</code>	Take pictures and videos
Characters	<code>org.gnome.Characters</code>	Character map application
Clocks	<code>org.gnome.clocks</code>	Keep track of time
Color Profile Viewer	<code>org.gnome.ColorViewer</code>	Inspect and compare installed color profiles
Connections	<code>org.gnome.Connections</code>	View and use other desktops
Contacts	<code>org.gnome.Contacts</code>	Manage your contacts
Disk Usage Analyzer	<code>org.gnome.baobab</code>	Check folder sizes and available disk space
Document Scanner	<code>org.gnome.SimpleScan</code>	Make a digital copy of your photos and documents
Document Viewer	<code>org.gnome.Evince</code>	Document viewer for popular document formats
Extensions	<code>org.gnome.Extensions</code>	Manage your GNOME Extensions
Fonts	<code>org.gnome.font-viewer</code>	View fonts on your system
Image Viewer	<code>org.gnome.Loupe</code>	View images
Logs	<code>org.gnome.Logs</code>	View detailed event logs for the system
Maps	<code>org.gnome.Maps</code>	Find places around the world
Music	<code>org.gnome.Music</code>	Play and organize your music collection
Text Editor	<code>org.gnome.TextEditor</code>	Edit text files
Videos	<code>org.gnome.Totem</code>	Play movies
Weather	<code>org.gnome.Weather</code>	Show weather conditions and forecast

Name	ID	Description
Web	org.gnome.Epiphany	Browse the web

Internet

Name	ID	Description
Eolie	org.gnome.Eolie	Web browser
Evolution	org.gnome.Evolution	Manage your email, contacts and schedule
Fractal	org.gnome.Fractal	Chat on Matrix
Geary	org.gnome.Geary	Send and receive email
Polari	org.gnome.Polari	Talk to people on IRC

Multimedia

Name	ID	Description
Cheese	org.gnome.Cheese	Take photos and videos with your webcam, with fun graphical effects
Decibels	org.gnome.Decibels	Play audio files
EasyTAG	org.gnome.EasyTAG	Edit audio file metadata
Eye of GNOME	org.gnome.eog	Browse and rotate images
gThumb Image Viewer	org.gnome.gThumb	View and organize your images
Identity	org.gnome.gitlab.YaLTeR.Identity	Compare images and videos
Lollypop	org.gnome.Lollypop	Play and organize your music collection
Photos	org.gnome.Photos	Access, organize and share your photos on GNOME
Podcasts	org.gnome.Podcasts	Listen to your favorite shows
Rhythmbox	org.gnome.Rhythmbox3	Play and organize all your music
Shotwell	org.gnome.Shotwell	Digital photo organizer
Showtime	org.gnome.Showtime	Watch without distraction
Sound Juicer	org.gnome.SoundJuicer	CD ripper with a clean interface and simple preferences

Name	ID	Description
Sound Recorder	<code>org.gnome.SoundRecorder</code>	A simple, modern sound recorder for GNOME
Video Trimmer	<code>org.gnome.gitlab.YaLTeR.VideoTrimmer</code>	Trim videos quickly

Productivity

Name	ID	Description
Apostrophe	<code>org.gnome.gitlab.somas.Apostrophe</code>	Edit Markdown in style
Bookup	<code>org.gnome.gitlab.ilhooq.Bookup</code>	Streamline notes with Markdown!
Break Timer	<code>org.gnome.BreakTimer</code>	Computer break reminders for GNOME
Citations	<code>org.gnome.World.Citations</code>	Manage your bibliography
Endeavour	<code>org.gnome.TODO</code>	Manage your tasks
Fava	<code>org.gnome.gitlab.johannesjh.favagtk</code>	Do your finances using fava and beancount
Getting Things GNOME!	<code>org.gnome.GTG</code>	Personal tasks and TODO-list items organizer
Gnote	<code>org.gnome.Gnote</code>	A simple note-taking application
Hamster	<code>org.gnome.Hamster</code>	Personal time keeping tool
Iotas	<code>org.gnome.World.Iotas</code>	Simple note taking
Notes	<code>org.gnome.Notes</code>	Notes for GNOME
Papers	<code>org.gnome.Papers</code>	Read documents
Pinpoint	<code>org.gnome.Pinpoint</code>	Excellent presentations for hackers
Pulp	<code>org.gnome.gitlab.cheywood.Pulp</code>	Skim excessive feeds
Recipes	<code>org.gnome.Recipes</code>	GNOME loves to cook
Solanum	<code>org.gnome.Solanum</code>	Balance working time and break time
Translation Editor	<code>org.gnome.Gtranslator</code>	Translate and localize applications and libraries

Games

Name	ID	Description
Aisleriot Solitaire	<code>org.gnome.Aisleriot</code>	Play many different solitaire games

Name	ID	Description
GNOME Chess	<code>org.gnome.Chess</code>	Play the classic two-player board game of chess
Crossword Editor	<code>org.gnome.Crosswords.Editor</code>	Create crossword puzzles
Crosswords	<code>org.gnome.Crosswords</code>	Solve crossword puzzles
Four-in-a-row	<code>org.gnome.Four-in-a-row</code>	Make lines of the same color to win
HexGL	<code>org.gnome.HexGL</code>	Space racing game
Hitori	<code>org.gnome.Hitori</code>	Play the Hitori puzzle game
GNOME Klotski	<code>org.gnome.Klotski</code>	Slide blocks to solve the puzzle
Lights Off	<code>org.gnome.LightsOff</code>	Turn off all the lights
Mahjongg	<code>org.gnome.Mahjongg</code>	Match tiles and clear the board
GNOME Mines	<code>org.gnome.Mines</code>	Clear hidden mines from a minefield
Nibbles	<code>org.gnome.Nibbles</code>	Guide a worm around a maze
Quadrappel	<code>org.gnome.Quadrappel</code>	Fit falling blocks together
Reversi	<code>org.gnome.Reversi</code>	Dominate the board in a classic reversi game, or play the reversed variant
GNOME Robots	<code>org.gnome.Robots</code>	Avoid the robots and make them crash into each other
GNOME Sudoku	<code>org.gnome.Sudoku</code>	Test yourself in the classic puzzle
Swell Foop	<code>org.gnome.SwellFoop</code>	Clear the screen by removing groups of colored and shaped tiles
Tali	<code>org.gnome.Tali</code>	Roll dice and score points
GNOME Taquin	<code>org.gnome.Taquin</code>	Slide tiles to their correct places
GNOME Tetravex	<code>org.gnome.Tetravex</code>	Reorder tiles to fit a square
GNOME 2048	<code>org.gnome.TwentyFortyEight</code>	Obtain the 2048 tile
Atomix	<code>org.gnome.atomix</code>	Build molecules out of single atoms
Five or More	<code>org.gnome.five-or-more</code>	Remove colored balls from the board by forming lines
gbrainy	<code>org.gnome.gbrainy</code>	gbrainy is a game to train memory, arithmetical, verbal and logical skills.
Convolution	<code>org.gnome.gitlab.bazylevnik0.Convolution</code>	Maze escaping game

Tools

Name	ID	Description
Brasero	<code>org.gnome.Brasero</code>	Create and copy CDs and DVDs
Buffer	<code>org.gnome.gitlab.cheywood.Buffer</code>	Embrace ephemeral text
Cowsay	<code>org.gnome.gitlab.Cowsay</code>	State of the art Cowsay generator
Déjà Dup Backups	<code>org.gnome.DejaDup</code>	Protect yourself from data loss
File Roller	<code>org.gnome.FileRoller</code>	Open, modify and create compressed archive files
Firmware	<code>org.gnome.Firmware</code>	Install firmware on devices
gedit	<code>org.gnome.gedit</code>	Text editor
GMetronome	<code>org.gnome.gitlab.dqpb.GMetronome</code>	Maintain a steady tempo
GNOME Network Displays	<code>org.gnome.NetworkDisplays</code>	Screencasting for GNOME
Keysign	<code>org.gnome.Keysign</code>	OpenPGP Keysigning helper
Passwords and Keys	<code>org.gnome.seahorse.Application</code>	Manage your passwords and encryption keys
Pika Backup	<code>org.gnome.World.PikaBackup</code>	Keep your data safe
Secrets	<code>org.gnome.World.Secrets</code>	Manage your passwords
Sushi	<code>org.gnome.NautilusPreviewer</code>	Provide a facility for quickly viewing different kinds of files

Software development

Name	ID	Description
Boxes	<code>org.gnome.Boxes</code>	Virtualization made simple
Builder	<code>org.gnome.Builder</code>	Create applications for GNOME
D-Spy	<code>org.gnome.dsby</code>	Analyze D-Bus connections
Devhelp	<code>org.gnome.Devhelp</code>	A developer tool for browsing and searching API documentation
GHex	<code>org.gnome.GHex</code>	Inspect and edit binary files
gitg	<code>org.gnome.gitg</code>	Graphical user interface for git
Glade	<code>org.gnome.Glade</code>	Create or open user interface designs for GTK+ applications
Meld	<code>org.gnome.meld</code>	Compare and merge your files