

---

**Sunshine**

**ReenigneArcher**

**Mar 30, 2023**



# ABOUT

<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	About . . . . .	1
1.2	System Requirements . . . . .	1
1.3	Integrations . . . . .	2
1.4	Support . . . . .	2
1.5	Downloads . . . . .	2
1.6	Stats . . . . .	2
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Binaries . . . . .	3
2.2	Docker . . . . .	3
2.3	Linux . . . . .	3
2.4	macOS . . . . .	7
2.5	Windows . . . . .	8
<b>3</b>	<b>Docker</b>	<b>9</b>
3.1	Important note . . . . .	9
3.2	Build your own containers . . . . .	9
3.3	Where used . . . . .	10
3.4	Port and Volume mappings . . . . .	10
3.5	Supported Architectures . . . . .	11
<b>4</b>	<b>Third Party Packages</b>	<b>13</b>
4.1	AUR . . . . .	13
4.2	Chocolatey . . . . .	13
4.3	nixpkgs . . . . .	13
4.4	Scoop . . . . .	13
4.5	Solus . . . . .	13
4.6	Winget . . . . .	14
4.7	Legacy GitHub Repo . . . . .	14
<b>5</b>	<b>Usage</b>	<b>15</b>
5.1	Network . . . . .	16
5.2	Arguments . . . . .	16
5.3	Setup . . . . .	16
5.4	Shortcuts . . . . .	19
5.5	Application List . . . . .	19
5.6	Considerations . . . . .	20
5.7	HDR Support . . . . .	20
5.8	Tutorials . . . . .	21

<b>6</b>	<b>App Examples</b>	<b>23</b>
6.1	Common Examples . . . . .	23
6.2	Linux . . . . .	25
6.3	macOS . . . . .	26
6.4	Windows . . . . .	26
<b>7</b>	<b>Advanced Usage</b>	<b>27</b>
7.1	Performance Tips . . . . .	27
7.2	Configuration . . . . .	27
7.3	General . . . . .	28
7.4	Controls . . . . .	29
7.5	Display . . . . .	31
7.6	Audio . . . . .	34
7.7	Network . . . . .	36
7.8	Encoding . . . . .	38
7.9	Advanced . . . . .	49
<b>8</b>	<b>Changelog</b>	<b>51</b>
8.1	0.19.0 - 2023-03-29 . . . . .	51
8.2	0.18.4 - 2023-02-20 . . . . .	52
8.3	0.18.3 - 2023-02-13 . . . . .	52
8.4	0.18.2 - 2023-02-13 . . . . .	53
8.5	0.18.1 - 2023-01-31 . . . . .	53
8.6	0.18.0 - 2023-01-29 . . . . .	53
8.7	0.17.0 - 2023-01-08 . . . . .	54
8.8	0.16.0 - 2022-12-13 . . . . .	55
8.9	0.15.0 - 2022-10-30 . . . . .	56
8.10	0.14.1 - 2022-08-09 . . . . .	57
8.11	0.14.0 - 2022-06-15 . . . . .	57
8.12	0.13.0 - 2022-02-27 . . . . .	58
8.13	0.12.0 - 2022-02-13 . . . . .	58
8.14	0.11.1 - 2021-10-04 . . . . .	58
8.15	0.11.0 - 2021-10-04 . . . . .	58
8.16	0.10.1 - 2021-08-21 . . . . .	58
8.17	0.10.0 - 2021-08-20 . . . . .	59
8.18	0.9.0 - 2021-07-11 . . . . .	59
8.19	0.8.0 - 2021-06-30 . . . . .	59
8.20	0.7.7 - 2021-06-24 . . . . .	60
8.21	0.7.1 - 2021-06-18 . . . . .	60
8.22	0.7.0 - 2021-06-16 . . . . .	60
8.23	0.6.0 - 2021-05-26 . . . . .	60
8.24	0.5.0 - 2021-05-13 . . . . .	60
8.25	0.4.0 - 2020-05-03 . . . . .	61
8.26	0.3.1 - 2020-04-24 . . . . .	61
8.27	0.3.0 - 2020-04-23 . . . . .	61
8.28	0.2.0 - 2020-03-21 . . . . .	61
8.29	0.1.1 - 2020-01-30 . . . . .	61
8.30	0.1.0 - 2020-01-27 . . . . .	62
<b>9</b>	<b>GameStream</b>	<b>63</b>
9.1	Migration . . . . .	63
9.2	Limitations . . . . .	63
<b>10</b>	<b>General</b>	<b>65</b>
10.1	Forgotten Credentials . . . . .	65

10.2	Web UI Access . . . . .	65
10.3	Nvidia issues . . . . .	65
<b>11</b>	<b>Linux</b>	<b>67</b>
11.1	KMS Streaming fails . . . . .	67
<b>12</b>	<b>macOS</b>	<b>69</b>
12.1	Dynamic session lookup failed . . . . .	69
<b>13</b>	<b>Windows</b>	<b>71</b>
13.1	No gamepad detected . . . . .	71
<b>14</b>	<b>Build</b>	<b>73</b>
14.1	Building Locally . . . . .	73
14.2	Remote Build . . . . .	73
<b>15</b>	<b>Linux</b>	<b>75</b>
15.1	Requirements . . . . .	75
15.2	CUDA . . . . .	78
15.3	npm dependencies . . . . .	78
15.4	Build . . . . .	79
<b>16</b>	<b>macOS</b>	<b>81</b>
16.1	Requirements . . . . .	81
16.2	npm dependencies . . . . .	81
16.3	Build . . . . .	82
<b>17</b>	<b>Windows</b>	<b>83</b>
17.1	Requirements . . . . .	83
17.2	npm dependencies . . . . .	83
17.3	Build . . . . .	83
<b>18</b>	<b>Contributing</b>	<b>85</b>
<b>19</b>	<b>Localization</b>	<b>87</b>
19.1	CrowdIn . . . . .	88
19.2	Extraction . . . . .	88
<b>20</b>	<b>Testing</b>	<b>91</b>
20.1	Clang Format . . . . .	91
20.2	Sphinx . . . . .	91
20.3	Unit Testing . . . . .	92
<b>21</b>	<b>Legal</b>	<b>93</b>
21.1	Commercial Use . . . . .	93
<b>22</b>	<b>src</b>	<b>95</b>
22.1	Example Documentation Blocks . . . . .	95
22.2	Code . . . . .	96
	<b>Index</b>	<b>135</b>



## OVERVIEW

LizardByte has the full documentation hosted on [Read the Docs](#).

## 1.1 About

Sunshine is a self-hosted game stream host for Moonlight. Offering low latency, cloud gaming server capabilities with support for AMD, Intel, and Nvidia GPUs for hardware encoding. Software encoding is also available. You can connect to Sunshine from any Moonlight client on a variety of devices. A web UI is provided to allow configuration, and client pairing, from your favorite web browser. Pair from the local server or any mobile device.

## 1.2 System Requirements

**Warning:** This table is a work in progress. Do not purchase hardware based on this.

### Minimum Requirements

GPU	AMD: VCE 1.0 or higher, see <a href="#">obs-amd hardware support</a> Intel: VA-API-compatible, see: <a href="#">VA-API hardware support</a> Nvidia: NVENC enabled cards, see <a href="#">nvenc support matrix</a>
CPU	AMD: Ryzen 3 or higher Intel: Core i3 or higher
RAM	4GB or more
OS	Windows: 10+ (Windows Server not supported) macOS: 11.7+ Linux/Debian: 11 (bullseye) Linux/Fedora: 36+ Linux/Ubuntu: 20.04+ (focal)
Network	Host: 5GHz, 802.11ac Client: 5GHz, 802.11ac

### 4k Suggestions

GPU	AMD: Video Coding Engine 3.1 or higher
	Intel: HD Graphics 510 or higher
	Nvidia: GeForce GTX 1080 or higher
CPU	AMD: Ryzen 5 or higher
	Intel: Core i5 or higher
Network	Host: CAT5e ethernet or better
	Client: CAT5e ethernet or better

### **HDR Suggestions**

GPU	AMD: Video Coding Engine 3.4 or higher
	Intel: UHD Graphics 730 or higher
	Nvidia: Pascal-based GPU (GTX 10-series) or higher
CPU	AMD: todo
	Intel: todo
Network	Host: CAT5e ethernet or better
	Client: CAT5e ethernet or better

## **1.3 Integrations**

## **1.4 Support**

Our support methods are listed in our [LizardByte Docs](#).

## **1.5 Downloads**

## **1.6 Stats**



## INSTALLATION

The recommended method for running Sunshine is to use the *binaries* bundled with the *latest release*.

**Attention:** Additional setup is required after installation. See *Setup*.

### 2.1 Binaries

Binaries of Sunshine are created for each release. They are available for Linux, macOS, and Windows. Binaries can be found in the *latest release*.

---

**Tip:** Some third party packages also exist. See *Third Party Packages*.

---

### 2.2 Docker

Docker images are available on [Dockerhub.io](https://hub.docker.com/r/sunshine-sdr/sunshine) and [ghcr.io](https://github.com/sunshine-sdr/ghcr.io).

See *Docker* for additional information.

### 2.3 Linux

Follow the instructions for your preferred package type below.

#### CUDA Compatibility

CUDA is used for NVFBC capture.

---

**Tip:** See [CUDA GPUS](#) to cross reference Compute Capability to your GPU.

---

Package	CUDA Version	Min Driver	CUDA Compute Capabilities
PKGBUILD	User dependent	User dependent	User dependent
sunshine.AppImage	11.8.0	450.80.02	50;52;60;61;62;70;75;80;86;90;35
sunshine.pkg.tar.zst	11.8.0	450.80.02	50;52;60;61;62;70;75;80;86;90;35
sunshine_{arch}.flatpak	12.0.0	525.60.13	50;52;60;61;62;70;75;80;86;90
sunshine-debian-bullseye-{arch}.deb	11.8.0	450.80.02	50;52;60;61;62;70;75;80;86;90;35
sunshine-fedora-36-{arch}.rpm	12.0.0	525.60.13	50;52;60;61;62;70;75;80;86;90
sunshine-fedora-37-{arch}.rpm	12.0.0	525.60.13	50;52;60;61;62;70;75;80;86;90
sunshine-ubuntu-20.04-{arch}.deb	11.8.0	450.80.02	50;52;60;61;62;70;75;80;86;90;35
sunshine-ubuntu-22.04-{arch}.deb	11.8.0	450.80.02	50;52;60;61;62;70;75;80;86;90;35

### 2.3.1 AppImage

According to AppImageLint the supported distro matrix of the AppImage is below.

- [×] Debian oldstable (buster)
- [✓] Debian stable (bullseye)
- [✓] Debian testing (bookworm)
- [✓] Debian unstable (sid)
- [✓] Ubuntu kinetic
- [✓] Ubuntu jammy
- [✓] Ubuntu focal
- [×] Ubuntu bionic
- [×] Ubuntu xenial
- [×] Ubuntu trusty
- [×] CentOS 7

1. Download `sunshine.AppImage` to your home directory.

2. Open terminal and run the following code.

```
./sunshine.AppImage --install
```

**Start:**

```
./sunshine.AppImage --install && ./sunshine.AppImage
```

**Uninstall:**

```
./sunshine.AppImage --remove
```

### 2.3.2 Archlinux PKGBUILD

1. Open terminal and run the following code.

```
wget https://github.com/LizardByte/Sunshine/releases/latest/download/PKGBUILD
makepkg -fi
```

**Uninstall:**

```
pacman -R sunshine
```

### 2.3.3 Archlinux pkg

1. Open terminal and run the following code.

```
wget https://github.com/LizardByte/Sunshine/releases/latest/download/sunshine.pkg.
↪tar.zst
pacman -U --noconfirm sunshine.pkg.tar.zst
```

**Uninstall:**

```
pacman -R sunshine
```

### 2.3.4 Debian Package

1. Download sunshine-{ubuntu-version}.deb and run the following code.

```
sudo apt install -f ./sunshine-{ubuntu-version}.deb
```

---

**Note:** The {ubuntu-version} is the version of ubuntu we built the package on. If you are not using Ubuntu and have an issue with one package, you can try another.

---

---

**Tip:** You can double click the deb file to see details about the package and begin installation.

---

**Uninstall:**

```
sudo apt remove sunshine
```

### 2.3.5 Flatpak Package

1. Install [Flatpak](#) as required.
2. Download sunshine\_{arch}.flatpak and run the following code.

---

**Note:** Be sure to replace {arch} with the architecture for your operating system.

---

**System level (recommended)**

```
flatpak install --system ./sunshine_{arch}.flatpak
```

### User level

```
flatpak install --user ./sunshine_{arch}.flatpak
```

### Additional installation (required)

```
flatpak run --command=additional-install.sh dev.lizardbyte.sunshine
```

### Start:

#### X11 and NVFBC capture (X11 Only)

```
flatpak run dev.lizardbyte.sunshine
```

#### KMS capture (Wayland & X11)

```
sudo -i PULSE_SERVER=unix:${pactl info | awk '/Server String/{print$3}')}  
↪ flatpak run dev.lizardbyte.sunshine
```

### Uninstall:

```
flatpak run --command=remove-additional-install.sh dev.lizardbyte.sunshine  
flatpak uninstall --delete-data dev.lizardbyte.sunshine
```

## 2.3.6 RPM Package

1. Add *rpmfusion* repositories by running the following code.

```
sudo dnf install https://mirrors.rpmfusion.org/free/fedora/rpmfusion-free-release-  
↪ ${rpm -E %fedora}.noarch.rpm \  
https://mirrors.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-${rpm -E  
↪ %fedora}.noarch.rpm
```

2. Download *sunshine.rpm* and run the following code.

```
sudo dnf install ./sunshine.rpm
```

---

**Tip:** You can double click the rpm file to see details about the package and begin installation.

---

### Uninstall:

```
sudo dnf remove sunshine
```

## 2.4 macOS

Sunshine on macOS is experimental. Gamepads do not work. Other features may not work as expected.

### 2.4.1 pkg

**Warning:** The *pkg* does not include runtime dependencies.

1. Download the `sunshine.pkg` file and install it as normal.

**Uninstall:**

```
cd /etc/sunshine/assets
uninstall_pkg.sh
```

### 2.4.2 Portfile

1. Install [MacPorts](#)
2. Update the Macports sources.

```
sudo nano /opt/local/etc/macports/sources.conf
```

**Add this line, replacing your username, below the line that starts with `rsync`.**

```
file:///Users/<username>/ports
```

Ctrl+x, then Y to exit and save changes.

3. Download the Portfile to `~/Downloads` and run the following code.

```
mkdir -p ~/ports/multimedia/sunshine
mv ~/Downloads/Portfile ~/ports/multimedia/sunshine/
cd ~/ports
portindex
sudo port install sunshine
```

4. The first time you start Sunshine, you will be asked to grant access to screen recording and your microphone.

**Uninstall:**

```
sudo port uninstall sunshine
```

## 2.5 Windows

### 2.5.1 Installer

1. Download and install `sunshine-windows-installer.exe`

**Attention:** You should carefully select or unselect the options you want to install. Do not blindly install or enable features.

To uninstall, find Sunshine in the list [here](#) and select “Uninstall” from the overflow menu. Different versions of Windows may provide slightly different steps for uninstall.

### 2.5.2 Standalone

1. Download and extract `sunshine-windows-portable.zip`

To uninstall, delete the extracted directory which contains the `sunshine.exe` file.

## 3.1 Important note

Starting with v0.18.0, tag names have changed. You may no longer use `latest`, `master`, `vX.X.X`.

## 3.2 Build your own containers

This image provides a method for you to easily use the latest Sunshine release in your own docker projects. It is not intended to use as a standalone container at this point, and should be considered experimental.

```
ARG SUNSHINE_VERSION=latest
ARG SUNSHINE_OS=ubuntu-22.04
FROM lizardbyte/sunshine:${SUNSHINE_VERSION}-${SUNSHINE_OS}

# install Steam, Wayland, etc.

ENTRYPOINT steam && sunshine
```

### 3.2.1 SUNSHINE\_VERSION

- `latest`, `master`, `vX.X.X`
- `nightly`
- commit hash

### 3.2.2 SUNSHINE\_OS

Sunshine images are available with the following tag suffixes, based on their respective base images.

- `archlinux`
- `debian-bullseye`
- `fedora-36`
- `fedora-37`
- `ubuntu-20.04`
- `ubuntu-22.04`

### 3.2.3 Tags

You must combine the `SUNSHINE_VERSION` and `SUNSHINE_OS` to determine the tag to pull. The format should be `<SUNSHINE_VERSION>-<SUNSHINE_OS>`. For example, `latest-ubuntu-22.04`.

See all our available tags on [docker hub](#) or [ghcr](#) for more info.

## 3.3 Where used

This is a list of docker projects using Sunshine. Something missing? Let us know about it!

- [Games on Whales](#)

## 3.4 Port and Volume mappings

Examples are below of the required mappings. The configuration file will be saved to `/config` in the container.

### 3.4.1 Using docker run

Create and run the container (substitute your `<values>`):

```
docker run -d \
  --name=<image_name> \
  --restart=unless-stopped
  -e PUID=<uid> \
  -e PGID=<gid> \
  -e TZ=<timezone> \
  -v <path to data>:/config \
  -p 47984-47990:47984-47990/tcp \
  -p 48010:48010 \
  -p 47998-48000:47998-48000/udp \
  <image>
```

### 3.4.2 Using docker-compose

Create a `docker-compose.yml` file with the following contents (substitute your `<values>`):

```
version: '3'
services:
  <image_name>:
    image: <image>
    container_name: sunshine
    restart: unless-stopped
    volumes:
      - <path to data>:/config
    environment:
      - PUID=<uid>
      - PGID=<gid>
      - TZ=<timezone>
```

(continues on next page)



(continued from previous page)

```
ports:
- "47984-47990:47984-47990/tcp"
- "48010:48010"
- "47998-48000:47998-48000/udp"
```

### 3.4.3 Parameters

You must substitute the <values> with your own settings.

Parameters are split into two halves separated by a colon. The left side represents the host and the right side the container.

**Example:** `-p external:internal` - This shows the port mapping from internal to external of the container. Therefore `-p 47990:47990` would expose port 47990 from inside the container to be accessible from the host's IP on port 47990 (e.g. `http://<host_ip>:47990`). The internal port must be 47990, but the external port may be changed (e.g. `-p 8080:47990`). All the ports listed in the `docker run` and `docker-compose` examples are required.

Parameter	Function	Example Value	Required
<code>-p &lt;port&gt;:47990</code>	Web UI Port	47990	True
<code>-v &lt;path to data&gt;:/config</code>	Volume mapping	/home/sunshine	True
<code>-e PUID=&lt;uid&gt;</code>	User ID	1001	False
<code>-e PGID=&lt;gid&gt;</code>	Group ID	1001	False
<code>-e TZ=&lt;timezone&gt;</code>	Lookup TZ value	America/New_York	False

#### User / Group Identifiers:

When using data volumes (-v flags) permissions issues can arise between the host OS and the container. To avoid this issue you can specify the user PUID and group PGID. Ensure the data volume directory on the host is owned by the same user you specify.

In this instance PUID=1001 and PGID=1001. To find yours use `id` user as below:

```
$ id dockeruser
uid=1001(dockeruser) gid=1001(dockergroup) groups=1001(dockergroup)
```

If you want to change the PUID or PGID after the image has been built, it will require rebuilding the image.

## 3.5 Supported Architectures

Specifying `lizardbyte/sunshine:latest-<SUNSHINE_OS>` or `ghcr.io/lizardbyte/sunshine:latest-<SUNSHINE_OS>` should retrieve the correct image for your architecture.

The architectures supported by these images are shown in the table below.

tag suffix	amd64/x86_64	arm64/aarch64
archlinux		
debian-bullseye		
fedora-36		
fedora-37		
ubuntu-20.04		
ubuntu-22.04		

## THIRD PARTY PACKAGES

<b>Danger:</b> These packages are not maintained by LizardByte. Use at your own risk.
---------------------------------------------------------------------------------------

### 4.1 AUR

### 4.2 Chocolatey

### 4.3 nixpkgs

### 4.4 Scoop

### 4.5 Solus

## 4.6 Winget

## 4.7 Legacy GitHub Repo

**Attention:** This repo is not maintained. Thank you to Loki for bringing this amazing project to life!

## USAGE

1. See the [setup](#) section for your specific OS.
2. If you did not install the service, then start sunshine with the following command, unless a start command is listed in the specified package [installation](#) instructions.

---

**Note:** A service is a process that runs in the background. Running multiple instances of Sunshine is not advised.

---

### Basic usage

```
sunshine
```

### Specify config file

```
sunshine <directory of conf file>/sunshine.conf
```

---

**Note:** You do not need to specify a config file. If no config file is entered the default location will be used.

---

**Attention:** The configuration file specified will be created if it doesn't exist.

3. Configure Sunshine in the web ui

The web ui is available on <https://localhost:47990> by default. You may replace *localhost* with your internal ip address.

**Attention:** Ignore any warning given by your browser about “insecure website”. This is due to the SSL certificate being self signed.

**Caution:** If running for the first time, make sure to note the username and password that you created.

### Add games and applications.

This can be configured in the web ui.

---

**Note:** Additionally, apps can be configured manually. *src\_assets/<os>/config/apps.json* is an example of a list of applications that are started just before running a stream. This is the directory within the GitHub

repo.

---

4. In Moonlight, you may need to add the PC manually.
5. When Moonlight request you insert the correct pin on sunshine:
  - Login to the web ui
  - Go to “PIN” in the Navbar
  - Type in your PIN and press Enter, you should get a Success Message
  - In Moonlight, select one of the Applications listed

## 5.1 Network

The Sunshine user interface will be available on port 47990 by default.

**Warning:** Exposing ports to the internet can be dangerous. Do this at your own risk.

## 5.2 Arguments

To get a list of available arguments run the following:

```
sunshine --help
```

## 5.3 Setup

### 5.3.1 Linux

The *deb*, *rpm*, *Flatpak* and *AppImage* packages handle these steps automatically. Third party packages may not.

Sunshine needs access to *uinput* to create mouse and gamepad events.

1. Add user to group *input*, if this is the first time installing.

```
sudo usermod -a -G input $USER
```

2. Create *udev* rules.

```
echo 'KERNEL=="uinput", GROUP="input", MODE="0660", OPTIONS+="static_node=uinput
→ "' | \
sudo tee /etc/udev/rules.d/85-sunshine-input.rules
```

3. Optionally, configure autostart service
  - filename: `~/.config/systemd/user/sunshine.service`
  - contents:

```
[Unit]
Description=Sunshine self-hosted game stream host for Moonlight.
StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
ExecStart=<see table>
Restart=on-failure
RestartSec=5s
#Flatpak Only
#ExecStop=flatpak kill dev.lizardbyte.sunshine

[Install]
WantedBy=graphical-session.target
```

package	ExecStart	Auto Configured
aur	/usr/bin/sunshine	✓
deb	/usr/bin/sunshine	✓
rpm	/usr/bin/sunshine	✓
AppImage	~/sunshine.AppImage	✓
Flatpak	flatpak run dev.lizardbyte.sunshine	✓

**Start once**

```
systemctl --user start sunshine
```

**Start on boot**

```
systemctl --user enable sunshine
```

**4. Additional Setup for KMS**

**Note:** cap\_sys\_admin may as well be root, except you don't need to be root to run it. It is necessary to allow Sunshine to use KMS.

**Enable**

```
sudo setcap cap_sys_admin+p $(readlink -f $(which sunshine))
```

**Disable (for Xorg/X11)**

```
sudo setcap -r $(readlink -f $(which sunshine))
```

**5. Reboot**

```
sudo reboot now
```

### 5.3.2 macOS

Sunshine can only access microphones on macOS due to system limitations. To stream system audio use [Soundflower](#) or [BlackHole](#).

---

**Note:** Command Keys are not forwarded by Moonlight. Right Option-Key is mapped to CMD-Key.

---

**Caution:** Gamepads are not currently supported.

#### Configure autostart service

##### MacPorts

```
sudo port load Sunshine
```

### 5.3.3 Windows

For gamepad support, install [ViGEmBus](#)

#### Sunshine firewall

##### Add rule

```
cd /d "C:\Program Files\Sunshine\scripts"  
add-firewall-rule.bat
```

##### Remove rule

```
cd /d "C:\Program Files\Sunshine\scripts"  
remove-firewall-rule.bat
```

#### Sunshine service

##### Enable

```
cd /d "C:\Program Files\Sunshine\scripts"  
install-service.bat
```

##### Disable

```
cd /d "C:\Program Files\Sunshine\scripts"  
uninstall-service.bat
```



## 5.4 Shortcuts

All shortcuts start with CTRL + ALT + SHIFT, just like Moonlight

- CTRL + ALT + SHIFT + N - Hide/Unhide the cursor (This may be useful for Remote Desktop Mode for Moonlight)
- CTRL + ALT + SHIFT + F1/F12 - Switch to different monitor for Streaming

## 5.5 Application List

- Applications should be configured via the web UI.
- A basic understanding of working directories and commands is required.
- You can use Environment variables in place of values
- \$(HOME) will be replaced by the value of \$HOME
- \$\$ will be replaced by \$, e.g. \$\$ (HOME) will become \$(HOME)
- env - Adds or overwrites Environment variables for the commands/applications run by Sunshine
- "Variable name": "Variable value"
- apps - The list of applications
- Advanced users may want to edit the application list manually. The format is json.
- **Example json application:**

```
{
  "cmd": "command to open app",
  "detached": [
    "some-command",
    "another-command"
  ],
  "image-path": "/full-path/to/png-image",
  "name": "An App",
  "output": "/full-path/to/command-log-file",
  "prep-cmd": [
    {
      "do": "some-command",
      "undo": "undo-that-command"
    }
  ],
  "working-dir": "/full-path/to/working-directory"
}
```

- cmd - The main application
- detached - A list of commands to be run and forgotten about
  - \* If not specified, a process is started that sleeps indefinitely
- image-path - The full path to the cover art image to use.
- name - The name of the application/game
- output - The file where the output of the command is stored

- `prep-cmd` - A list of commands to be run before/after the application
    - \* If any of the prep-commands fail, starting the application is aborted
    - \* `do` - Run before the application
      - If it fails, all `undo` commands of the previously succeeded `do` commands are run
    - \* `undo` - Run after the application has terminated
      - Failures of `undo` commands are ignored
  - `working-dir` - The working directory to use. If not specified, Sunshine will use the application directory.
- For more examples see [app examples](#).

## 5.6 Considerations

- When an application is started, if there is an application already running, it will be terminated.
- When the application has been shutdown, the stream shuts down as well.
  - For example, if you attempt to run `steam` as a `cmd` instead of `detached` the stream will immediately fail. This is due to the method in which the steam process is executed. Other applications may behave similarly.
- The “Desktop” app works the same as any other application except it has no commands. It does not start an application, instead it simply starts a stream. If you removed it and would like to get it back, just add a new application with the name “Desktop” and “desktop.png” as the image path.
- For the Linux flatpak you must prepend commands with `flatpak-spawn --host`.

## 5.7 HDR Support

Streaming HDR content is supported for Windows hosts with NVIDIA, AMD, or Intel GPUs that support encoding HEVC Main 10. You must have an HDR-capable display or EDID emulator dongle connected to your host PC to activate HDR in Windows.

- Ensure you enable the HDR option in your Moonlight client settings, otherwise the stream will be SDR.
- A good HDR experience relies on proper HDR display calibration both in Windows and in game. HDR calibration can differ significantly between client and host displays.
- We recommend calibrating the display by streaming the Windows HDR Calibration app to your client device and saving an HDR calibration profile to use while streaming.
- You may also need to tune the brightness slider or HDR calibration options in game to the different HDR brightness capabilities of your client’s display.
- Older games that use NVIDIA-specific NVAPI HDR rather than native Windows 10 OS HDR support may not display in HDR.
- Some GPUs can produce lower image quality or encoding performance when streaming in HDR compared to SDR.

## 5.8 Tutorials

Tutorial videos are available [here](#).

---

### Community!

Tutorials are community generated. Want to contribute? Reach out to us on our discord server.

---



## APP EXAMPLES

Since not all applications behave the same, we decided to create some examples to help you get started adding games and applications to Sunshine.

**Attention:** Throughout these examples, any fields not shown are left blank. You can enhance your experience by adding an image or a log file (via the **Output** field).

### 6.1 Common Examples

#### 6.1.1 Desktop

Field	Value
Application Name	Desktop
Image	desktop.png

#### 6.1.2 Steam Big Picture

**Note:** Steam is launched as a detached command because Steam starts with a process that self updates itself and the original process is killed. Since the original process ends it will not work as a regular command.

Field	Linux	macOS	Windows
Application Name	Steam Big Picture		
Detached Commands	setsid steam	steam://open/bigpicture	steam steam://open/bigpicture
Image	steam.png		

### 6.1.3 Epic Game Store game

---

**Note:** Using URI method will be the most consistent between various games, but does not allow a game to be launched using the “Command” and therefore the stream will not end when the game ends.

---

#### URI (Epic)

Field	Windows
Application Name	Surviving Mars
Detached Commands	cmd /C "start com.epicgames.launcher://apps/d759128018124dcabb1fbee9bb28e178%3A20729b9176c2"

#### Binary (Epic w/ working directory)

Field	Windows
Application Name	Surviving Mars
Command	cmd /c "MarsEpic.exe"
Working Directory	C:\Program Files\Epic Games\SurvivingMars

#### Binary (Epic w/o working directory)

Field	Windows
Application Name	Surviving Mars
Command	"C:\Program Files\Epic Games\SurvivingMars\MarsEpic.exe"

### 6.1.4 Steam game

---

**Note:** Using URI method will be the most consistent between various games, but does not allow a game to be launched using the “Command” and therefore the stream will not end when the game ends.

---

**URI (Steam)**

Field	Linux	macOS	Windows
Application Name	Surviving Mars		
Detached Commands	setsid steam steam://rungameid/464920	open steam://rungameid/464920	cmd /C "start steam://rungameid/464920"

**Binary (Steam w/ working directory)**

Field	Linux	macOS	Windows
Application Name	Surviving Mars		
Command	MarsSteam		cmd /c "MarsSteam.exe"
Working Directory	~/.steam/steam/SteamApps/common/Surviving Mars		C:\Program Files (x86)\Steam\steamapps\common\Surviving Mars

**Binary (Steam w/o working directory)**

Field	Linux	macOS	Windows
Application Name	Surviving Mars		
Command	~/.steam/steam/SteamApps/common/Surviving Mars/MarsSteam		"C:\Program Files (x86)\Steam\steamapps\common\Surviving Mars\MarsSteam.exe"

## 6.2 Linux

### 6.2.1 Changing Resolution and Refresh Rate (Linux - X11)

Field	Value
Command Preparations	Do: xrandr --output HDMI-1 --mode 1920x1080 --rate 60 Undo: xrandr --output HDMI-1 --mode 3840x2160 --rate 120

### 6.2.2 Changing Resolution and Refresh Rate (Linux - Wayland)

Field	Value
Command Preparations	Do: wlr-xrandr --output HDMI-1 --mode 1920x1080@60Hz Undo: wlr-xrandr --output HDMI-1 --mode 3840x2160@120Hz

## 6.2.3 Flatpak

**Attention:** Because Flatpak packages run in a sandboxed environment and do not normally have access to the host, the Flatpak of Sunshine requires commands to be prefixed with `flatpak-spawn --host`.

## 6.3 macOS

### 6.3.1 Changing Resolution and Refresh Rate (macOS)

---

**Note:** This example uses the *displayplacer* tool to change the resolution. This tool can be installed following instructions in their [GitHub repository](#).

---

Field	Value
Command Preparations	Do: <code>displayplacer "id:&lt;screenId&gt; res:1920x1080 hz:60 scaling:on origin:(0,0) degree:0"</code>
	Undo: <code>displayplacer "id:&lt;screenId&gt; res:3840x2160 hz:120 scaling:on origin:(0,0) degree:0"</code>

---

## 6.4 Windows

### 6.4.1 Changing Resolution and Refresh Rate (Windows)

---

**Note:** This example uses the *QRes* tool to change the resolution and refresh rate. This tool can be downloaded from their [SourceForge repository](#).

---

Field	Value
Command Preparations	Do: <code>FullPath\qres.exe /x:1920 /y:1080 /r:60</code>
	Undo: <code>FullPath\qres.exe /x:3840 /y:2160 /r:120</code>

---

---

**Tip:** You can change your host resolution to match the client resolution automatically using the [Nonary/ResolutionAutomation](#) project.

---



## ADVANCED USAGE

Sunshine will work with the default settings for most users. In some cases you may want to configure Sunshine further.

### 7.1 Performance Tips

#### 7.1.1 AMD

In Windows, enabling *Enhanced Sync* in AMD's settings may help reduce the latency by an additional frame. This applies to *amfenc* and *libx264*.

#### 7.1.2 Nvidia

Enabling *Fast Sync* in Nvidia settings may help reduce latency.

### 7.2 Configuration

The default location for the configuration file is listed below. You can use another location if you choose, by passing in the full configuration file path as the first argument when you start Sunshine.

The default location of the `apps.json` is the same as the configuration file. You can use a custom location by modifying the configuration file.

#### Default File Location

Value	Description
Docker	/config/
Linux	~/.config/sunshine/
macOS	~/.config/sunshine/
Windows	%ProgramFiles%\Sunshine\config

#### Example

```
sunshine ~/sunshine_config.conf
```

To manually configure sunshine you may edit the `conf` file in a text editor. Use the examples as reference.

---

**Hint:** Some settings are not available within the web ui.

---

## 7.3 General

### 7.3.1 sunshine\_name

**Description**

The name displayed by Moonlight

**Default**

PC hostname

**Example**

```
sunshine_name = Sunshine
```

### 7.3.2 min\_log\_level

**Description**

The minimum log level printed to standard out.

**Choices**

Value	Description
verbose	verbose logging
debug	debug logging
info	info logging
warning	warning logging
error	error logging
fatal	fatal logging
none	no logging

**Default**

info

**Example**

```
min_log_level = info
```

### 7.3.3 log\_path

**Description**

The path where the sunshine log is stored.

**Default**

sunshine.log

**Example**

```
log_path = sunshine.log
```

### 7.3.4 global\_prep\_cmd

#### Description

A list of commands to be run before/after all applications. If any of the prep-commands fail, starting the application is aborted.

#### Default

[]

#### Example

```
global_prep_cmd = [{"do": "nircmd.exe setdisplay 1280 720 32 144", "undo": "nircmd.exe ↵
↵ setdisplay 2560 1440 32 144"}]
```

## 7.4 Controls

### 7.4.1 gamepad

#### Description

The type of gamepad to emulate on the host.

**Caution:** Applies to Windows only.

#### Choices

Value	Description
x360	xbox 360 controller
ds4	dualshock controller (PS4)

#### Default

x360

#### Example

```
gamepad = x360
```

### 7.4.2 back\_button\_timeout

#### Description

If, after the timeout, the back/select button is still pressed down, Home/Guide button press is emulated.

On Nvidia Shield, the home and power button are not passed to Moonlight.

**Tip:** If back\_button\_timeout < 0, then the Home/Guide button will not be emulated.

#### Default

2000

#### Example

```
back_button_timeout = 2000
```

### 7.4.3 key\_repeat\_delay

**Description**

The initial delay, in milliseconds, before repeating keys. Controls how fast keys will repeat themselves.

**Default**

500

**Example**

```
key_repeat_delay = 500
```

### 7.4.4 key\_repeat\_frequency

**Description**

How often keys repeat every second.

---

**Tip:** This configurable option supports decimals.

---

**Default**

24.9

**Example**

```
key_repeat_frequency = 24.9
```

### 7.4.5 keybindings

**Description**

Sometimes it may be useful to map keybindings. Wayland won't allow clients to capture the Win Key for example.

---

**Tip:** See [virtual key codes](#)

---

---

**Hint:** keybindings needs to have a multiple of two elements.

---

**Default**

```
0x10, 0xA0,  
0x11, 0xA2,  
0x12, 0xA4
```

**Example**

```
keybindings = [
    0x10, 0xA0,
    0x11, 0xA2,
    0x12, 0xA4,
    0x4A, 0x4B
]
```

### 7.4.6 key\_rightalt\_to\_key\_win

#### Description

It may be possible that you cannot send the Windows Key from Moonlight directly. In those cases it may be useful to make Sunshine think the Right Alt key is the Windows key.

#### Default

disabled

#### Example

```
key_rightalt_to_key_win = enabled
```

## 7.5 Display

### 7.5.1 adapter\_name

#### Description

Select the video card you want to stream.

---

**Tip:** To find the name of the appropriate values follow these instructions.

#### Linux + VA-API

Unlike with *amdvce* and *nvenc*, it doesn't matter if video encoding is done on a different GPU.

```
ls /dev/dri/renderD* # to find all devices capable of VAAPI

# replace ``renderD129`` with the device from above to lists the name and
↳ capabilities of the device
vainfo --display drm --device /dev/dri/renderD129 | \
  grep -E "((VAProfileH264High|VAProfileHEVCMain|VAProfileHEVCMain10).
↳ *VAEntrypointEncSlice)|Driver version"
```

To be supported by Sunshine, it needs to have at the very minimum: VAProfileH264High : VAEntrypointEncSlice

---

**Todo:** macOS

#### Windows

```
tools\dxgi-info.exe
```

### Default

Sunshine will select the default video card.

### Examples

#### Linux

```
adapter_name = /dev/dri/renderD128
```

---

**Todo:** macOS

---

#### Windows

```
adapter_name = Radeon RX 580 Series
```

## 7.5.2 output\_name

### Description

Select the display number you want to stream.

---

**Tip:** To find the name of the appropriate values follow these instructions.

#### Linux

During Sunshine startup, you should see the list of detected monitors:

```
Info: Detecting connected monitors
Info: Detected monitor 0: DVI-D-0, connected: false
Info: Detected monitor 1: HDMI-0, connected: true
Info: Detected monitor 2: DP-0, connected: true
Info: Detected monitor 3: DP-1, connected: false
Info: Detected monitor 4: DVI-D-1, connected: false
```

You need to use the value before the colon in the output, e.g. 1.

---

**Todo:** macOS

---

#### Windows

```
tools\dxgi-info.exe
```

---

### Default

Sunshine will select the default display.

### Examples

#### Linux

```
output_name = 0
```

---

**Todo:** macOS

---

**Windows**

```
output_name = \\.\DISPLAY1
```

**7.5.3 fps****Description**

The fps modes advertised by Sunshine.

---

**Note:** Some versions of Moonlight, such as Moonlight-nx (Switch), rely on this list to ensure that the requested fps is supported.

---

**Default**

```
[10, 30, 60, 90, 120]
```

**Example**

```
fps = [10, 30, 60, 90, 120]
```

**7.5.4 resolutions****Description**

The resolutions advertised by Sunshine.

---

**Note:** Some versions of Moonlight, such as Moonlight-nx (Switch), rely on this list to ensure that the requested resolution is supported.

---

**Default**

```
[
  352x240,
  480x360,
  858x480,
  1280x720,
  1920x1080,
  2560x1080,
  3440x1440,
  1920x1200,
  3860x2160,
  3840x1600,
]
```

**Example**

```
resolutions = [
  352x240,
  480x360,
  858x480,
  1280x720,
```

(continues on next page)

(continued from previous page)

```
1920x1080,  
2560x1080,  
3440x1440,  
1920x1200,  
3860x2160,  
3840x1600,  
]
```

## 7.5.5 dwmflush

### Description

Invoke `DwmFlush()` to sync screen capture to the Windows presentation interval.

**Caution:** Applies to Windows only. Alleviates visual stuttering during mouse movement. If enabled, this feature will automatically deactivate if the client framerate exceeds the host monitor's current refresh rate.

**Note:** If you disable this option, you may see video stuttering during mouse movement in certain scenarios. It is recommended to leave enabled when possible.

### Default

enabled

### Example

```
dwmflush = enabled
```

## 7.6 Audio

### 7.6.1 audio\_sink

#### Description

The name of the audio sink used for audio loopback.

**Tip:** To find the name of the audio sink follow these instructions.

#### Linux + pulseaudio

```
pacmd list-sinks | grep "name:"
```

#### Linux + pipewire

```
pactl info | grep Source  
# in some cases you'd need to use the `Sink` device, if `Source` doesn't work, so  
↪ try:  
pactl info | grep Sink
```



**macOS**

Sunshine can only access microphones on macOS due to system limitations. To stream system audio use [Soundflower](#) or [BlackHole](#).

**Windows**

```
tools\audio-info.exe
```

**Tip:** If you want to mute the host speakers, use [virtual\\_sink](#) instead.

**Default**

Sunshine will select the default audio device.

**Examples****Linux**

```
audio_sink = alsa_output.pci-0000_09_00.3.analog-stereo
```

**macOS**

```
audio_sink = BlackHole 2ch
```

**Windows**

```
audio_sink = {0.0.0.00000000}. {FD47D9CC-4218-4135-9CE2-0C195C87405B}
```

## 7.6.2 virtual\_sink

**Description**

The audio device that's virtual, like Steam Streaming Speakers. This allows Sunshine to stream audio, while muting the speakers.

**Tip:** See [audio\\_sink](#)!

**Tip:** These are some options for virtual sound devices.

- Stream Streaming Speakers (Linux, macOS, Windows)
  - To use this option, you must have Steam installed and have used Stream remote play at least once.
- [Virtual Audio Cable](#) (macOS, Windows)

**Example**

```
virtual_sink = {0.0.0.00000000}. {8edba70c-1125-467c-b89c-15da389bcd4}
```

## 7.7 Network

### 7.7.1 external\_ip

**Description**

If no external IP address is given, Sunshine will attempt to automatically detect external ip-address.

**Default**

Automatic

**Example**

```
external_ip = 123.456.789.12
```

### 7.7.2 port

**Description**

Set the family of ports used by Sunshine. Changing this value will offset other ports per the table below.

Port Description	Default Port	Difference from config port
HTTPS	47984 TCP	-5
HTTP	47989 TCP	0
Web	47990 TCP	+1
RTSP	48010 TCP	+21
Video	47998 UDP	+9
Control	47999 UDP	+10
Audio	48000 UDP	+11
Mic (unused)	48002 UDP	+13

**Attention:** Custom ports may not be supported by all Moonlight clients.

**Default**

47989

**Example**

```
port = 47989
```

### 7.7.3 pkey

**Description**

The private key. This must be 2048 bits.

**Default**

credentials/cakey.pem

**Example**

```
pkey = /dir/pkey.pem
```

## 7.7.4 cert

### Description

The certificate. Must be signed with a 2048 bit key.

### Default

credentials/cacert.pem

### Example

```
cert = /dir/cert.pem
```

## 7.7.5 origin\_pin\_allowed

### Description

The origin of the remote endpoint address that is not denied for HTTP method /pin.

### Choices

Value	Description
pc	Only localhost may access /pin
lan	Only LAN devices may access /pin
wan	Anyone may access /pin

### Default

pc

### Example

```
origin_pin_allowed = pc
```

## 7.7.6 origin\_web\_ui\_allowed

### Description

The origin of the remote endpoint address that is not denied for HTTPS Web UI.

### Choices

Value	Description
pc	Only localhost may access the web ui
lan	Only LAN devices may access the web ui
wan	Anyone may access the web ui

### Default

lan

### Example

```
origin_web_ui_allowed = lan
```

### 7.7.7 upnp

**Description**

Sunshine will attempt to open ports for streaming over the internet.

**Choices**

Value	Description
on	enable UPnP
off	disable UPnP

**Default**

disabled

**Example**

```
upnp = on
```

### 7.7.8 ping\_timeout

**Description**

How long to wait, in milliseconds, for data from Moonlight before shutting down the stream.

**Default**

10000

**Example**

```
ping_timeout = 10000
```

## 7.8 Encoding

### 7.8.1 channels

**Description**

This will generate distinct video streams, unlike simply broadcasting to multiple Clients.

When multicasting, it could be useful to have different configurations for each connected Client.

For instance:

- Clients connected through WAN and LAN have different bitrate constraints.
- Decoders may require different settings for color.

**Warning:** CPU usage increases for each distinct video stream generated.

**Default**

1

**Example**

```
channels = 1
```

### 7.8.2 fec\_percentage

#### Description

Percentage of error correcting packets per data packet in each video frame.

**Warning:** Higher values can correct for more network packet loss, but at the cost of increasing bandwidth usage.

#### Default

20

#### Range

1-255

#### Example

```
fec_percentage = 20
```

### 7.8.3 qp

#### Description

Quantization Parameter. Some devices don't support Constant Bit Rate. For those devices, QP is used instead.

**Warning:** Higher value means more compression, but less quality.

#### Default

28

#### Example

```
qp = 28
```

### 7.8.4 min\_threads

#### Description

Minimum number of threads used for software encoding.

**Note:** Increasing the value slightly reduces encoding efficiency, but the tradeoff is usually worth it to gain the use of more CPU cores for encoding. The ideal value is the lowest value that can reliably encode at your desired streaming settings on your hardware.

#### Default

1

#### Example

```
min_threads = 1
```

### 7.8.5 hevc\_mode

#### Description

Allows the client to request HEVC Main or HEVC Main10 video streams.

**Warning:** HEVC is more CPU-intensive to encode, so enabling this may reduce performance when using software encoding.

#### Choices

Value	Description
0	advertise support for HEVC based on encoder
1	do not advertise support for HEVC
2	advertise support for HEVC Main profile
3	advertise support for HEVC Main and Main10 (HDR) profiles

#### Default

0

#### Example

```
hevc_mode = 2
```

### 7.8.6 capture

#### Description

Force specific screen capture method.

**Caution:** Applies to Linux only.

#### Choices

Value	Description
nvfb	Use NVIDIA Frame Buffer Capture to capture direct to GPU memory. This is usually the fastest method for NVIDIA cards. For GeForce cards it will only work with drivers patched with <a href="#">nvidia-patch</a> or <a href="#">nvllax</a> .
wlr	Capture for wlroots based Wayland compositors via DMA-BUF.
kms	DRM/KMS screen capture from the kernel. This requires that sunshine has <code>cap_sys_admin</code> capability. See <a href="#">Linux Setup</a> .
x11	Uses XCB. This is the slowest and most CPU intensive so should be avoided if possible.

#### Default

Automatic. Sunshine will use the first capture method available in the order of the table above.

#### Example

```
capture = kms
```

### 7.8.7 encoder

#### Description

Force a specific encoder.

#### Choices

Value	Description
nvenc	For NVIDIA graphics cards
quicksync	For Intel graphics cards
amdvce	For AMD graphics cards
software	Encoding occurs on the CPU

#### Default

Sunshine will use the first encoder that is available.

#### Example

```
encoder = nvenc
```

### 7.8.8 sw\_preset

#### Description

The encoder preset to use.

---

**Note:** This option only applies when using software *encoder*.

---

---

**Note:** From [FFmpeg](#).

A preset is a collection of options that will provide a certain encoding speed to compression ratio. A slower preset will provide better compression (compression is quality per filesize). This means that, for example, if you target a certain file size or constant bit rate, you will achieve better quality with a slower preset. Similarly, for constant quality encoding, you will simply save bitrate by choosing a slower preset.

Use the slowest preset that you have patience for.

---

#### Choices

Value	Description
ultrafast	fastest
superfast	
veryfast	
faster	
fast	
medium	
slow	
slower	
veryslow	slowest

**Default**

superfast

**Example**

```
sw_preset = superfast
```

## 7.8.9 sw\_tune

**Description**

The tuning preset to use.

---

**Note:** This option only applies when using software *encoder*.

---

---

**Note:** From FFmpeg.

You can optionally use -tune to change settings based upon the specifics of your input.

---

**Choices**

Value	Description
film	use for high quality movie content; lowers deblocking
animation	good for cartoons; uses higher deblocking and more reference frames
grain	preserves the grain structure in old, grainy film material
stillimage	good for slideshow-like content
fastdecode	allows faster decoding by disabling certain filters
zerolatency	good for fast encoding and low-latency streaming

**Default**

zerolatency

**Example**

```
sw_tune = zerolatency
```



### 7.8.10 nv\_preset

#### Description

The encoder preset to use.

---

**Note:** This option only applies when using nvenc *encoder*. For more information on the presets, see [nvenc preset migration guide](#).

---

#### Choices

Value	Description
p1	fastest (lowest quality)
p2	faster (lower quality)
p3	fast (low quality)
p4	medium (default)
p5	slow (good quality)
p6	slower (better quality)
p7	slowest (best quality)

#### Default

p4

#### Example

```
nv_preset = p4
```

### 7.8.11 nv\_tune

#### Description

The encoder tuning profile.

---

**Note:** This option only applies when using nvenc *encoder*.

---

#### Choices

Value	Description
hq	high quality
ll	low latency
ull	ultra low latency
lossless	lossless

#### Default

ull

#### Example

```
nv_tune = ull
```

### 7.8.12 nv\_rc

**Description**

The encoder rate control.

---

**Note:** This option only applies when using nvenc *encoder*.

---

**Choices**

Value	Description
constqp	constant QP mode
vbr	variable bitrate
cbr	constant bitrate

**Default**

cbr

**Example**

```
nv_rc = cbr
```

### 7.8.13 nv\_coder

**Description**

The entropy encoding to use.

---

**Note:** This option only applies when using H264 with nvenc *encoder*.

---

**Choices**

Value	Description
auto	let ffmpeg decide
cabac	context adaptive binary arithmetic coding - higher quality
cavlc	context adaptive variable-length coding - faster decode

**Default**

auto

**Example**

```
nv_coder = auto
```

### 7.8.14 qsv\_preset

#### Description

The encoder preset to use.

---

**Note:** This option only applies when using quicksync *encoder*.

---

#### Choices

Value	Description
veryfast	fastest (lowest quality)
faster	faster (lower quality)
fast	fast (low quality)
medium	medium (default)
slow	slow (good quality)
slower	slower (better quality)
veryslow	slowest (best quality)

#### Default

medium

#### Example

```
qsv_preset = medium
```

### 7.8.15 qsv\_coder

#### Description

The entropy encoding to use.

---

**Note:** This option only applies when using H264 with quicksync *encoder*.

---

#### Choices

Value	Description
auto	let ffmpeg decide
cabac	context adaptive binary arithmetic coding - higher quality
cavlc	context adaptive variable-length coding - faster decode

#### Default

auto

#### Example

```
qsv_coder = auto
```

### 7.8.16 amd\_quality

**Description**

The encoder preset to use.

---

**Note:** This option only applies when using amdvce *encoder*.

---

**Choices**

Value	Description
speed	prefer speed
balanced	balanced
quality	prefer quality

**Default**

balanced

**Example**

```
amd_quality = balanced
```

### 7.8.17 amd\_rc

**Description**

The encoder rate control.

---

**Note:** This option only applies when using amdvce *encoder*.

---

**Choices**

Value	Description
cqp	constant qp mode
cbr	constant bitrate
vbr_latency	variable bitrate, latency constrained
vbr_peak	variable bitrate, peak constrained

**Default**

vbr\_latency

**Example**

```
amd_rc = vbr_latency
```

### 7.8.18 amd\_usage

**Description**

The encoder usage profile, used to balance latency with encoding quality.

---

**Note:** This option only applies when using amdvce *encoder*.

---

**Choices**

Value	Description
transcoding	transcoding (slowest)
webcam	webcam (slow)
lowlatency	low latency (fast)
ultralowlatency	ultra low latency (fastest)

**Default**

ultralowlatency

**Example**

```
amd_usage = ultralowlatency
```

### 7.8.19 amd\_prealysis

**Description**

Prealysis can increase encoding quality at the cost of latency.

---

**Note:** This option only applies when using amdvce *encoder*.

---

**Default**

disabled

**Example**

```
amd_prealysis = disabled
```

### 7.8.20 amd\_vbaq

**Description**

Variance Based Adaptive Quantization (VBAQ) can increase subjective visual quality.

---

**Note:** This option only applies when using amdvce *encoder*.

---

**Default**

enabled

**Example**

```
amd_vbaq = enabled
```

## 7.8.21 amd\_coder

### Description

The entropy encoding to use.

---

**Note:** This option only applies when using H264 with amdvce *encoder*.

---

### Choices

Value	Description
auto	let ffmpeg decide
cabac	context adaptive variable-length coding - higher quality
cavlc	context adaptive binary arithmetic coding - faster decode

### Default

auto

### Example

```
amd_coder = auto
```

## 7.8.22 vt\_software

### Description

Force Video Toolbox to use software encoding.

---

**Note:** This option only applies when using macOS.

---

### Choices

Value	Description
auto	let ffmpeg decide
disabled	disable software encoding
allowed	allow software encoding
forced	force software encoding

### Default

auto

### Example

```
vt_software = auto
```

### 7.8.23 vt\_realtime

**Description**

Realtime encoding.

---

**Note:** This option only applies when using macOS.

---

**Warning:** Disabling realtime encoding might result in a delayed frame encoding or frame drop.

**Default**

enabled

**Example**

```
vt_realtime = enabled
```

### 7.8.24 vt\_coder

**Description**

The entropy encoding to use.

---

**Note:** This option only applies when using macOS.

---

**Choices**

Value	Description
auto	let ffmpeg decide
cabac	
cavlc	

**Default**

auto

**Example**

```
vt_coder = auto
```

## 7.9 Advanced

### 7.9.1 file\_apps

**Description**

The application configuration file path. The file contains a json formatted list of applications that can be started by Moonlight.

**Default**

OS and package dependent

### Example

```
file_apps = apps.json
```

### 7.9.2 file\_state

#### Description

The file where current state of Sunshine is stored.

#### Default

sunshine\_state.json

### Example

```
file_state = sunshine_state.json
```

### 7.9.3 credentials\_file

#### Description

The file where user credentials for the UI are stored.

#### Default

sunshine\_state.json

### Example

```
credentials_file = sunshine_state.json
```



## CHANGELOG

### 8.1 0.19.0 - 2023-03-29

#### Breaking

- (Linux/Flatpak) Moved Flatpak to org.freedesktop.Platform 22.08 and Cuda 12.0.0 This will drop support for Nvidia GPUs with compute capability 3.5

#### Added

- (Input) Added option to suppress input from gamepads, keyboards, or mice
- (Input/Linux) Added unicode support for remote pasting (may not work on all DEs)
- (Input/Linux) Added XTest input fallback
- (UI) Added version notifications to web UI
- (Linux/Windows) Add system tray icon
- (Windows) Added ability to safely elevate commands that fail due to insufficient permissions when running as a service
- (Config) Added global prep commands, and ability to exclude an app from using global prep commands
- (Installer/Windows) Automatically install ViGEmBus if selected

#### Changed

- (Logging) Changed client verified messages to debug to prevent spamming the log
- (Config) Only save non default config values
- (Service/Linux) Use xdg-desktop-autostart for systemd service
- (Linux) Added config option to force capture method
- (Windows) Execute prep command in context of current user
- (Linux) Allow disconnected X11 outputs

#### Fixed

- (Input/Windows) Fix issue where internation keys were not translated correct, and modifier keys appeared stuck
- (Linux) Fixed startup when /dev/dri didn't exist
- (UI) Changes software encoding settings to select menu instead of text input
- (Initialization) Do not terminate upon failure, allowing access to the web UI

#### Dependencies

- Bump third-party/moonlight-common-c from 07beb0f to c9426a6
- Bump babel from 2.11.0 to 2.12.1
- Bump @fontawesome/fontawesome-free from 6.2.1 to 6.4.0
- Bump third-party/ViGEmClient from 9e842ba to 726404e
- Bump ffmpeg
- Bump third-party/miniupnp from 014c9df to e439318
- Bump furo from 2022.12.7 to 2023.3.27
- Bump third-party/nanors from 395e5ad to e9e242e

**Misc**

- (GitHub) Shared feature request board with Moonlight
- (Docs) Improved application examples
- (Docs) Added WIP documentation for source code using Doxygen and Breathe
- (Build) Fix linux clang build errors
- (Build/Archlinux) Skip irrelevant submodules
- (Build/Archlinux) Disable download timeout
- (Build/macOS) Support compiling for earlier releases of macOS
- (Docs) Add favicon
- (Docs) Add missing config default values
- (Build) Fix compiler warnings due to depreciated elements in C++17
- (Build) Fix libcurl link errors
- (Clang) Adjusted formatting rules

## **8.2 0.18.4 - 2023-02-20**

**Fixed**

- (Linux/AUR) Drop support of AUR package
- (Docker) General enhancements to docker images

## **8.3 0.18.3 - 2023-02-13**

**Added**

- (Linux) Added PKGBUILD for Archlinux based distros to releases
- (Linux) Added precompiled package for Archlinux based distros to releases
- (Docker) Added archlinux docker image (x86\_64 only)

## 8.4 0.18.2 - 2023-02-13

### Fixed

- (Video/KMV/Linux) Fixed wayland capture on Nvidia for KMS
- (Video/Linux) Implement vaSyncBuffer stuff for libva <2.9.0
- (UI) Fix issue where mime type was not being set for node\_modules when using a reverse proxy
- (UI/macOS) Added missing audio sink config options
- (Linux) Specify correct Boost dependency versions
- (Video/AMF) Add missing encoder tunables

## 8.5 0.18.1 - 2023-01-31

### Fixed

- (Linux) Fixed missing dependencies for deb and rpm packages
- (Linux) Use dynamic boost

## 8.6 0.18.0 - 2023-01-29

Attention, this release contains critical security fixes. Please update as soon as possible. Additionally, we are encouraging users to change your Sunshine password, especially if you expose the web UI (i.e. port 47790 by default) to the internet, or have ever uploaded your logs with verbose output to a public resource.

### Added

- (Windows) Add support for Intel QuickSync
- (Linux) Added aarch64 deb and rpm packages
- (Windows) Add support for hybrid graphics systems, such as laptops with both integrated and discrete GPUs
- (Linux) Add support for streaming from Steam Deck Gaming Mode
- (Windows) Add HDR support, see <https://docs.lizardbyte.dev/projects/sunshine/en/latest/about/usage.html#hdr-support>

### Fixed

- (Network) Refactor code for UPnP port forwarding
- (Video) Enforce 10 FPS encoding frame rate minimum to improve static image quality
- (Linux) deb and rpm packages are now specific to destination distro and version
- (Docs) Add nvidia/nvenc preset migration guide
- (Network) Performance optimizations
- (Video/Windows) Fix streaming to multiple clients from hardware encoder
- (Linux) Fix child process spawning
- (Security) Fix security vulnerability in implementation of SimpleWebServer
- (Misc) Rename “Steam BigPicture” to “Steam Big Picture” in default apps.json

- (Security) Scrub basic authorization header from logs
- (Linux) The systemd service will now restart in the event of a crash
- (Video/KMS/Linux) Fixed error: couldn't import RGB Image: 00003002 and 00003004
- (Video/Windows) Fix stream freezing triggered by the resolution changed
- (Installer/Windows) Fixes silent installation and other miscellaneous improvements
- (CPU) Significantly improved CPU usage

## **8.7 0.17.0 - 2023-01-08**

If you are running Sunshine as a service on Windows, we are strongly urging you to update to v0.17.0 as soon as possible. Older Windows versions of Sunshine had a security flaw in which the binary was located in a user-writable location which is problematic when running as a service or on a multi-user system. Additionally, when running Sunshine as a service, games and applications were launched as SYSTEM. This could lead to issues with save files and other game settings. In v0.17.0, games now run under your user account without elevated privileges.

### **Breaking**

- (Apps) Removed automatic desktop entry (Re-add by adding an empty application named “Desktop” with no commands, “desktop.png” can be added as the image.)
- (Windows) Improved user upgrade experience (Suggest to manually uninstall existing Sunshine version before this upgrade. Do NOT select to remove everything, if prompted. Make a backup of config files before uninstall.)
- (Windows) Move config files to specific directory (files will be migrated automatically if using Windows installer)
- (Dependencies) Fix npm path (breaking change for package maintainers)

### **Added**

- (macOS) Added initial support for arm64 on macOS through Macports portfile
- (Input) Added support for foreign keyboard input
- (Misc) Logs inside the WebUI and log to file
- (UI/Windows) Added an Apply button to configuration page when running as a service
- (Input/Windows) Enable Mouse Keys while streaming for systems with no physical mouse

### **Fixed**

- (Video) Improved capture performance
- (Audio) Improved audio bitrate and quality handling
- (Apps/Windows) Fixed PATH environment variable handling
- (Apps/Windows) Use the proper environment variable for the Program Files (x86) folder
- (Service/Windows) Fix SunshineSvc hanging if an error occurs during startup
- (Service/Windows) Spawn Sunshine.exe in a job object, so it is terminated if SunshineSvc.exe dies
- (Video) windows/vram: fix fringing in NV12 colour conversion
- (Apps/Windows) Launch games under the correct user account
- (Video) nvenc, amdvc: rework all user presets/options
- (Network) Generate certificates with unique serial numbers

- (Service/Windows) Graceful termination on shutdown, logoff, and service stop
- (Apps/Windows) Fix launching apps when Sunshine is running as admin
- (Misc) Remove/fix calls to std::abort()
- (Misc) Remove prompt to press enter after Sunshine exits
- (Misc) Make log priority consistent for execution messages
- (Apps) Applications in Moonlight clients are now updated automatically after editing
- (Video/Linux) Fix wayland capture on nvidia
- (Audio) Fix 7.1 surround channel mapping
- (Video) Fix NVENC profile values not applying
- (Network) Fix origin\_web\_ui\_allowed binding
- (Service/Windows) Self terminate/restart service if process hangs for 10 seconds
- (Input/Windows) Fix Windows masked cursor blending with GPU encoders
- (Video) Color conversion fixes and BT.2020 support

#### Dependencies

- Bump ffmpeg from 4.4 to 5.1
- ffmpeg\_patches: add amfenc delay/buffering fix
- CBS moved to ffmpeg submodules
- Migrate to upstream Simple-Web-Server submodule
- Bump third-party/TPCircularBuffer from bce9170 to 8833b3a
- Bump third-party/moonlight-common-c from 8169a31 to ef9ad52
- Bump third-party/miniupnp from 6f848ae to 207cf44
- Bump third-party/ViGEmClient from f719a1d to 9e842ba
- Bump bootstrap from 5.0.0 to 5.2.3
- Bump @fontawesome/fontawesome-free from 6.2.0 to 6.2.1

## 8.8 0.16.0 - 2022-12-13

#### Added

- Add cover finder
- (Docker) Add arm64 docker image
- (Flatpak) Add installation helper scripts
- (Windows) Add support for Unicode input messages

#### Fixed

- (Linux) Reintroduce Ubuntu 20.04 and 22.04 specific deb packages
- (Linux) Fixed udev and systemd file locations

#### Dependencies

- Bump babel from 2.10.3 to 2.11.0
- Bump sphinx-copybutton from 0.5.0 to 0.5.1
- Bump KSXGitHub/github-actions-deploy-aur from 2.5.0 to 2.6.0
- Use npm for web dependencies (breaking change for third-party package maintainers)
- Update moonlight-common-c
- Use pre-built ffmpeg from LizardByte/build-deps for all sunshine builds (breaking change for third-party package maintainers)
- Bump furo from 2022.9.29 to 2022.12.7

**Misc**

- Misc org level workflow updates
- Fix misc typos in docs
- Fix winget release

## **8.9 0.15.0 - 2022-10-30**

**Added**

- (Windows) Add firewall rules scripts
- (Windows) Automatically add and remove firewall rules at install/uninstall
- (Windows) Automatically add and remove service at install/uninstall
- (Docker) Official image added
- (Linux) Add aarch64 flatpak package

**Changed**

- (Windows/Linux/MacOS) - Move default config and apps file to assets directory
- (MacOS) Bump boost to 1.80 for macport builds
- (Linux) Remove backup and restore of config files

**Fixed**

- (Linux) - Create sunshine config directory if it doesn't exist
- (Linux) Remove portable home and config directories for AppImage
- (Windows) Include service install and uninstall scripts again
- (Windows) Automatically delete start menu entry upon uninstall
- (Windows) Automatically delete program install directory upon uninstall, with user prompt
- (Linux) Handle the case of no default audio sink
- (Windows/Linux/MacOS) Fix default image paths
- (Linux) Fix CUDA RGBA to NV12 conversion

## 8.10 0.14.1 - 2022-08-09

### Added

- (Linux) Flatpak package added
- (Linux) AUR package automated updates
- (Windows) Winget package automated updates

### Changed

- (General) Moved repo to @LizardByte GitHub org
- (WebUI) Fixed button spacing on home page
- (WebUI) Added Discord WidgetBot Crate

### Fixed

- (Linux/Mac) Default config and app files now copied to user home directory
- (Windows) Default config and app files now copied to working directory

## 8.11 0.14.0 - 2022-06-15

### Added

- (Documentation) Added Sphinx documentation available at <https://sunshinestream.readthedocs.io/en/latest/>
- (Development) Initial support for Localization
- (Linux) Add rpm package as release asset
- (macOS) Add Portfile as release asset
- (Windows) Add DwmFlush() call to improve capture
- (Windows) Add Windows installer

### Fixed

- (AMD) Fixed hwdevice being destroyed before context
- (Linux) Added missing dependencies to AppImage
- (Linux) Fixed rumble events causing game to freeze
- (Linux) Improved Pulse/Pipewire compatibility
- (Linux) Moved to single deb package
- (macOS) Fixed missing TPCircularBuffer submodule
- (Stream) Properly catch exceptions in stream broadcast handlers
- (Stream/Video) AVPacket fix

## 8.12 0.13.0 - 2022-02-27

### Added

- (macOS) Initial support for macOS (#40)

## 8.13 0.12.0 - 2022-02-13

### Added

- New command line argument `--version`
- Custom png poster support

### Changed

- Correct software bitrate calculation
- Increase vbv-bufsize to 1/10 of requested bitrate
- Improvements to Web UI

## 8.14 0.11.1 - 2021-10-04

### Changed

- (Linux) Fix search path for config file and assets

## 8.15 0.11.0 - 2021-10-04

### Added

- (Linux) Added support for wlroots based compositors on Wayland.
- (Windows) Added an icon for the executable

### Changed

- Fixed a bug causing segfault when connecting multiple controllers.
- (Linux) Improved NVENC, it now offloads converting images from RGB to NV12
- (Linux) Fixed a bug causes stuttering

## 8.16 0.10.1 - 2021-08-21

### Changed

- (Linux) Re-enabled KMS



## 8.17 0.10.0 - 2021-08-20

### Added

- Added support for Rumble with gamepads.
- Added support for keyboard shortcuts <— See the README for details.
- (Windows) A very basic script has been added in Sunshine-Windowstools <— This will start Sunshine at boot with the highest privileges which is needed to display the login prompt.

### Changed

- Some cosmetic changes to the WebUI.
- The first time the WebUI is opened, it will request the creation of a username/password pair from the user.
- Fixed audio crackling introduced in version 0.8.0
- (Linux) VA-API hardware encoding now works on Intel i7-6700 at least. <— For the best experience, using ffmpeg version 4.3 or higher is recommended.
- (Windows) Installing from debian package shouldn't overwrite your configuration files anymore. <— It's recommended that you back up /etc/sunshine/ before testing this.

## 8.18 0.9.0 - 2021-07-11

### Added

- Added audio encryption
- (Linux) Added basic NVENC support on Linux
- (Windows) The Windows version can now capture the lock screen and the UAC prompt as long as it's run through PsExec.exe <https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>

### Changed

- Sunshine will now accept expired or not-yet-valid certificates, as long as they are signed properly.
- Fixed compatibility with iOS version of Moonlight
- Drastically reduced chance of being forced to skip error correction due to video frame size
- (Linux) sunshine.service will be installed automatically.

## 8.19 0.8.0 - 2021-06-30

### Added

- Added mDNS support: Moonlight will automatically find Sunshine.
- Added UPnP support. It's off by default.

## 8.20 0.7.7 - 2021-06-24

### Added

- (Linux) Added installation package for Debian

### Changed

- Fixed incorrect scaling for absolute mouse coordinates when using multiple monitors.
- Fixed incorrect colors when scaling for software encoder

## 8.21 0.7.1 - 2021-06-18

### Changed

- (Linux) Fixed an issue where it was impossible to start sunshine on ubuntu 20.04

## 8.22 0.7.0 - 2021-06-16

### Added

- Added a Web Manager. Accessible through: <https://localhost:47990> or <https://:47990>
- (Linux) Added hardware encoding support for AMD on Linux

### Changed

- (Linux) Moved certificates and saved pairings generated during runtime to .config/sunshine on Linux

## 8.23 0.6.0 - 2021-05-26

### Added

- Added support for surround audio

### Changed

- Maintain aspect ratio when scaling video
- Fix issue where Sunshine is forced to drop frames when they are too large

## 8.24 0.5.0 - 2021-05-13

### Added

- Added support for absolute mouse coordinates
- (Linux) Added support for streaming specific monitor on Linux
- (Windows) Added support for AMF on Windows

## 8.25 0.4.0 - 2020-05-03

### Changed

- prep-cmd is now optional in apps.json
- Fixed bug causing video artifacts
- Fixed bug preventing Moonlight from closing app on exit
- Fixed bug causing preventing keyboard keys from repeating on latest version of Moonlight
- Fixed bug causing segfault when another session of sunshine was already running
- Fixed bug causing crash when monitor has resolution 1366x768

## 8.26 0.3.1 - 2020-04-24

### Changed

- Fix a memory leak.

## 8.27 0.3.0 - 2020-04-23

### Changed

- Hardware acceleration on NVidia GPU's for Video encoding on Windows

## 8.28 0.2.0 - 2020-03-21

### Changed

- Multicasting is now supported: You can set the maximum simultaneous connections with the configurable option: channels
- Configuration variables can be overwritten on the command line: "name=value" -> it can be useful to set min\_log\_level=debug without modifying the configuration file
- Switches to make testing the pairing mechanism more convenient has been added, see "sunshine -help" for details

## 8.29 0.1.1 - 2020-01-30

### Added

- (Linux) Added deb package and service for Linux

### 8.30 0.1.0 - 2020-01-27

#### Added

- The first official release for Sunshine!

## **GAMESTREAM**

Nvidia announced that their GameStream service for Nvidia Games clients will be discontinued in February 2023. Luckily, Sunshine performance is now on par with Nvidia GameStream. Many users have even reported that Sunshine outperforms GameStream, so rest assured that Sunshine will be equally performant moving forward.

### **9.1 Migration**

We have developed a simple migration tool to help you migrate your GameStream games and apps to Sunshine automatically. Please check out our [GSMS](#) project if you're interested in an automated migration option. At the time of writing this GSMS offers the ability to migrate your custom games and apps. The working directory, command, and image are all set in Sunshine's `apps.json` file. The box-art image is also copied to a specified directory.

### **9.2 Limitations**

Sunshine does have some limitations, as compared to Nvidia GameStream.

- Automatic game/application list.
- Changing game settings automatically, to optimize streaming.



## 10.1 Forgotten Credentials

If you forgot your credentials to the web UI, try this.

```
sunshine --creds <new username> <new password>
```

## 10.2 Web UI Access

Can't access the web UI?

1. Check firewall rules.

## 10.3 Nvidia issues

NvFBC, NvENC, or general issues with Nvidia graphics card.

- Consumer grade Nvidia cards are software limited to a specific number of encodes. See [Video Encode and Decode GPU Support Matrix](#) for more info.
- You can usually bypass the restriction with a driver patch. See Keylase's [Linux](#) or [Windows](#) patches for more guidance.





## 11.1 KMS Streaming fails

If screencasting fails with KMS, you may need to run the following to force unprivileged screencasting.

```
sudo setcap -r $(readlink -f $(which sunshine))
```



## 12.1 Dynamic session lookup failed

If you get this error:

*Dynamic session lookup supported but failed: launchd did not provide a socket path, verify that org.freedesktop.dbus-session.plist is loaded!*

Try this.

```
launchctl load -w /Library/LaunchAgents/org.freedesktop.dbus-session.plist
```



## **13.1 No gamepad detected**

1. Verify that you've installed [ViGEmBus](#).



## BUILD

Sunshine binaries are built using [CMake](#). Cross compilation is not supported. That means the binaries must be built on the target operating system and architecture.

### 14.1 Building Locally

#### 14.1.1 Clone

Ensure [git](#) is installed and run the following:

```
git clone https://github.com/lizardbyte/sunshine.git --recurse-submodules
cd sunshine && mkdir build && cd build
```

#### 14.1.2 Compile

See the section specific to your OS.

- *Linux*
- *macOS*
- *Windows*

### 14.2 Remote Build

It may be beneficial to build remotely in some cases. This will enable easier building on different operating systems.

1. Fork the project
2. Activate workflows
3. Trigger the *CI* workflow manually
4. Download the artifacts/binaries from the workflow run summary





## 15.1 Requirements

### 15.1.1 Debian Bullseye

End of Life: TBD

#### Install Requirements

```
sudo apt update && sudo apt install \  
    build-essential \  
    cmake \  
    libavdevice-dev \  
    libboost-filesystem-dev \  
    libboost-locale-dev \  
    libboost-log-dev \  
    libboost-program-options-dev \  
    libboost-thread-dev \  
    libcap-dev \ # KMS  
    libcurl4-openssl-dev \  
    libdrm-dev \ # KMS  
    libevdev-dev \  
    libmfx-dev \ # x86_64 only  
    libnuma-dev \  
    libopus-dev \  
    libpulse-dev \  
    libssl-dev \  
    libva-dev \  
    libvdpau-dev \  
    libwayland-dev \ # Wayland  
    libx11-dev \ # X11  
    libxcb-shm0-dev \ # X11  
    libxcb-xfixes0-dev \ # X11  
    libxcb1-dev \ # X11  
    libxfixes-dev \ # X11  
    libxrandr-dev \ # X11  
    libxtst-dev \ # X11  
    nodejs \  
    npm \  
    nvidia-cuda-dev \ # Cuda, NvFBC  
    nvidia-cuda-toolkit \ # Cuda, NvFBC
```

## 15.1.2 Fedora 36, 37

End of Life: TBD

### Install Requirements

```
sudo dnf update && \
sudo dnf group install "Development Tools" && \
sudo dnf install \
    boost-devel \
    cmake \
    gcc \
    gcc-c++ \
    intel-mediasdk-devel \ # x86_64 only
    libappindicator-gtk3-devel \
    libcap-devel \
    libcurl-devel \
    libdrm-devel \
    libevdev-devel \
    libva-devel \
    libvdpau-devel \
    libX11-devel \ # X11
    libxcb-devel \ # X11
    libXcursor-devel \ # X11
    libXfixes-devel \ # X11
    libXi-devel \ # X11
    libXinerama-devel \ # X11
    libXrandr-devel \ # X11
    libXtst-devel \ # X11
    mesa-libGL-devel \
    npm \
    numactl-devel \
    openssl-devel \
    opus-devel \
    pulseaudio-libs-devel \
    rpm-build \ # if you want to build an RPM binary package
    wget \ # necessary for cuda install with `run` file
    which \ # necessary for cuda install with `run` file
```

## 15.1.3 Ubuntu 20.04

End of Life: April 2030

### Install Requirements

```
sudo apt update && sudo apt install \
    build-essential \
    cmake \
    g++-10 \
    libappindicator3-dev \
    libavdevice-dev \
    libboost-filesystem-dev \
    libboost-locale-dev \
```

(continues on next page)

(continued from previous page)

```

libboost-log-dev \
libboost-thread-dev \
libboost-program-options-dev \
libcap-dev \ # KMS
libdrm-dev \ # KMS
libevdev-dev \
libmfx-dev \ # x86_64 only
libnuma-dev \
libopus-dev \
libpulse-dev \
libssl-dev \
libva-dev \
libvdpau-dev \
libwayland-dev \ # Wayland
libx11-dev \ # X11
libxcb-shm0-dev \ # X11
libxcb-xfixes0-dev \ # X11
libxcb1-dev \ # X11
libxf86-dev \ # X11
libxrandr-dev \ # X11
libxtst-dev \ # X11
nodejs \
npm \
wget # necessary for cuda install with `run` file

```

### Update gcc alias

```

update-alternatives --install \
/usr/bin/gcc gcc /usr/bin/gcc-10 100 \
--slave /usr/bin/g++ g++ /usr/bin/g++-10 \
--slave /usr/bin/gcov gcov /usr/bin/gcov-10 \
--slave /usr/bin/gcc-ar gcc-ar /usr/bin/gcc-ar-10 \
--slave /usr/bin/gcc-ranlib gcc-ranlib /usr/bin/gcc-ranlib-10

```

## 15.1.4 Ubuntu 22.04

End of Life: April 2027

### Install Requirements

```

sudo apt update && sudo apt install \
build-essential \
cmake \
libappindicator3-dev \
libavdevice-dev \
libboost-filesystem-dev \
libboost-locale-dev \
libboost-log-dev \
libboost-thread-dev \
libboost-program-options-dev \
libcap-dev \ # KMS

```

(continues on next page)

(continued from previous page)

```
libdrm-dev \ # KMS
libevdev-dev \
libmfx-dev \ # x86_64 only
libnuma-dev \
libopus-dev \
libpulse-dev \
libssl-dev \
libwayland-dev \ # Wayland
libx11-dev \ # X11
libxcb-shm0-dev \ # X11
libxcb-xfixes0-dev \ # X11
libxcb1-dev \ # X11
libxf86-dev \ # X11
libxrandr-dev \ # X11
libxtst-dev \ # X11
nodejs \
npm \
nvidia-cuda-dev \ # CUDA, NvFBC
nvidia-cuda-toolkit # CUDA, NvFBC
```

## 15.2 CUDA

If the version of CUDA available from your distro is not adequate, manually install CUDA.

**Tip:** The version of CUDA you use will determine compatibility with various GPU generations. See [CUDA compatibility](#) for more info.

Select the appropriate run file based on your desired CUDA version and architecture according to [CUDA Toolkit Archive](#).

```
wget https://developer.download.nvidia.com/compute/cuda/11.4.2/local_installers/cuda_11.
↪4.2_470.57.02_linux.run \
--progress=bar:force:noscroll -q --show-progress -O ./cuda.run
chmod a+x ./cuda.run
./cuda.run --silent --toolkit --toolkitpath=/usr --no-opengl-libs --no-man-page --no-drm
rm ./cuda.run
```

## 15.3 npm dependencies

Install npm dependencies.

```
npm install
```

## 15.4 Build

**Attention:** Ensure you are in the build directory created during the clone step earlier before continuing.

```
cmake ..  
make -j ${nproc}  
  
cpack -G DEB # optionally, create a deb package  
cpack -G RPM # optionally, create a rpm package
```



## 16.1 Requirements

macOS Big Sur and Xcode 12.5+

Use either [MacPorts](#) or [Homebrew](#)

### 16.1.1 MacPorts

#### Install Requirements

```
sudo port install avahi boost180 cmake curl libopus npm9 pkgconfig
```

### 16.1.2 Homebrew

#### Install Requirements

```
brew install boost cmake node opus  
# if there are issues with an SSL header that is not found:  
cd /usr/local/include  
ln -s ../opt/openssl/include/openssl .
```

## 16.2 npm dependencies

#### Install npm dependencies.

```
npm install
```

## 16.3 Build

**Attention:** Ensure you are in the build directory created during the clone step earlier before continuing.

```
cmake ..  
make -j ${nproc}  
  
cpack -G DragNDrop # optionally, create a macOS dmg package
```

**If cmake fails complaining to find Boost, try to set the path explicitly.**

```
cmake -DBOOST_ROOT=[boost path] .., e.g., cmake -DBOOST_ROOT=/opt/local/libexec/boost/1.  
80 ..
```



## 17.1 Requirements

First you need to install [MSYS2](#), then startup “MSYS2 MinGW 64-bit” and execute the following codes.

**Update all packages:**

```
pacman -Suy
```

**Install dependencies:**

```
pacman -S base-devel cmake diffutils gcc git make mingw-w64-x86_64-binutils \  
mingw-w64-x86_64-boost mingw-w64-x86_64-cmake mingw-w64-x86_64-curl \  
mingw-w64-x86_64-libmfx mingw-w64-x86_64-openssl mingw-w64-x86_64-opus \  
mingw-w64-x86_64-toolchain
```

## 17.2 npm dependencies

Install nodejs and npm. Downloads available [here](#).

**Install npm dependencies.**

```
npm install
```

## 17.3 Build

**Attention:** Ensure you are in the build directory created during the clone step earlier before continuing.

```
cmake -G "MinGW Makefiles" ..  
mingw32-make -j$(nproc)  
  
cpack -G NSIS # optionally, create a windows installer  
cpack -G ZIP # optionally, create a windows standalone package
```



## CONTRIBUTING

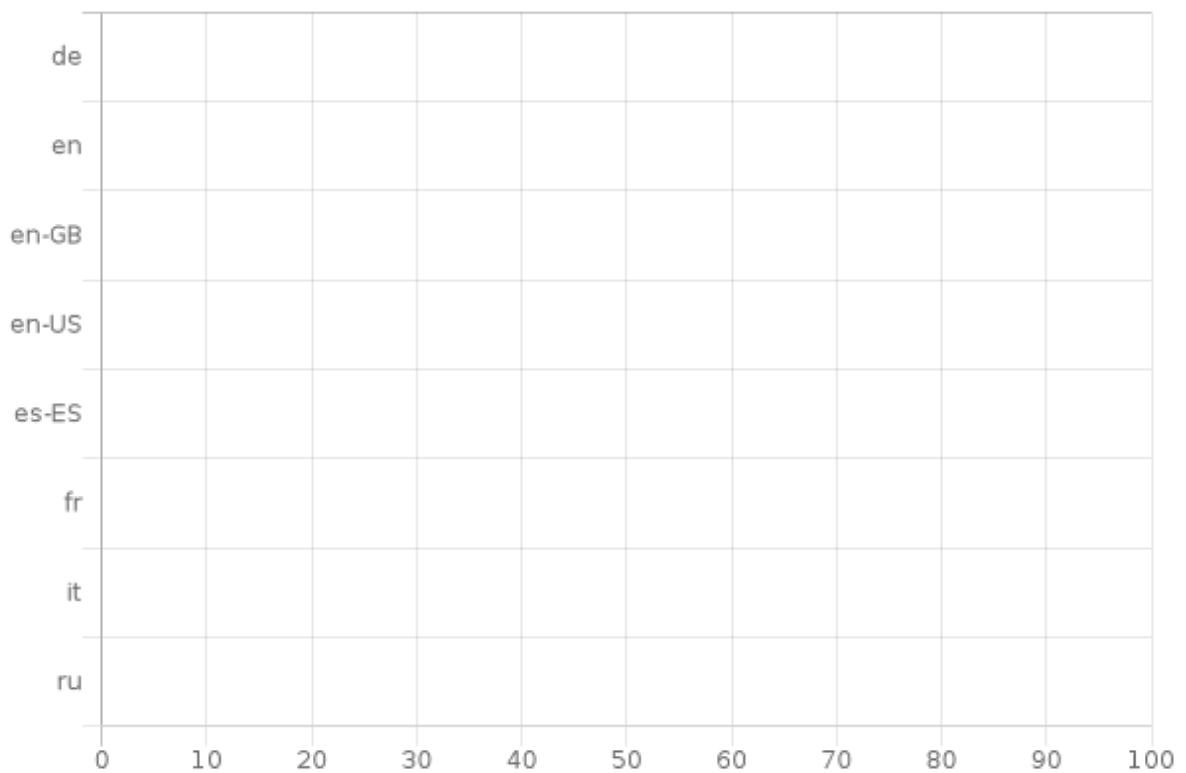
Read our contribution guide in our organization level [docs](#).



## LOCALIZATION

Sunshine is being localized into various languages. The default language is *en* (English) and is highlighted green.

Graph



## 19.1 CrowdIn

The translations occur on [CrowdIn](#). Feel free to contribute to localization there. Only elements of the API are planned to be translated.

**Attention:** The rest API has not yet been implemented.

### Translations Basics

- The brand names *LizardByte* and *Sunshine* should never be translated.
- Other brand names should never be translated. Examples:
  - AMD
  - Nvidia

### CrowdIn Integration

How does it work?

When a change is made to sunshine source code, a workflow generates new translation templates that get pushed to CrowdIn automatically.

When translations are updated on CrowdIn, a push gets made to the *l10n\_nightly* branch and a PR is made against the *nightly* branch. Once PR is merged, all updated translations are part of the project and will be included in the next release.

## 19.2 Extraction

There should be minimal cases where strings need to be extracted from source code; however it may be necessary in some situations. For example if a system tray icon is added it should be localized as it is user interfacing.

- **Wrap the string to be extracted in a function as shown.**

```
#include <boost/locale.hpp>
boost::locale::translate("Hello world!")
```

---

**Tip:** More examples can be found in the documentation for [boost locale](#).

---

**Warning:** This is for information only. Contributors should never include manually updated template files, or manually compiled language files in Pull Requests.

Strings are automatically extracted from the code to the *locale/sunshine.po* template file. The generated file is used by CrowdIn to generate language specific template files. The file is generated using the *.github/workflows/localize.yml* workflow and is run on any push event into the *nightly* branch. Jobs are only run if any of the following paths are modified.

```
- 'src/**'
```

When testing locally it may be desirable to manually extract, initialize, update, and compile strings. Python is required for this, along with the python dependencies in the *.scripts/requirements.txt* file. Additionally, [xgettext](#) must be installed.

**Extract, initialize, and update**

```
python ./scripts/_locale.py --extract --init --update
```

**Compile**

```
python ./scripts/_locale.py --compile
```





## 20.1 Clang Format

Source code is tested against the *.clang-format* file for linting errors. The workflow file responsible for clang format testing is *.github/workflows/cpp-clang-format-lint.yml*.

**Test clang-format locally.**

```
find ./ -iname *.cpp -o -iname *.h -iname *.m -iname *.mm | xargs clang-format -i
```

## 20.2 Sphinx

Sunshine uses [Sphinx](#) for documentation building. Sphinx, along with other required python dependencies are included in the *./docs/requirements.txt* file. Python is required to build sphinx docs. Installation and setup of python will not be covered here.

Doxygen is used to generate the XML files required by Sphinx. Doxygen can be obtained from [Doxygen downloads](#). Ensure that the *doxygen* executable is in your path.

The config file for Sphinx is *docs/source/conf.py*. This is already included in the repo and should not be modified.

The config file for Doxygen is *docs/Doxyfile*. This is already included in the repo and should not be modified.

**Test with Sphinx**

```
cd docs  
make html
```

Alternatively

```
cd docs  
sphinx-build -b html source build
```

## 20.3 Unit Testing

---

**Todo:** Sunshine does not currently have any unit tests. If you would like to help us improve please get in contact with us, or make a PR with suggested changes.

---

**Attention:** This documentation is for informational purposes only and is not intended as legal advice. If you have any legal questions or concerns about using Sunshine, we recommend consulting with a lawyer.

Sunshine is licensed under the GPL-3.0 license, which allows for free use and modification of the software. The full text of the license can be reviewed [here](#).

## 21.1 Commercial Use

Sunshine can be used in commercial applications without any limitations. This means that businesses and organizations can use Sunshine to create and sell products or services without needing to seek permission or pay a fee.

However, it is important to note that the GPL-3.0 license does not grant any rights to distribute or sell the encoders contained within Sunshine. If you plan to sell access to Sunshine as part of their distribution, you are responsible for obtaining the necessary licenses to do so. This may include obtaining a license from the Motion Picture Experts Group (MPEG-LA) and/or any other necessary licensing requirements.

In summary, while Sunshine is free to use, it is the user's responsibility to ensure compliance with all applicable licensing requirements when redistributing the software as part of a commercial offering. If you have any questions or concerns about using Sunshine in a commercial setting, we recommend consulting with a lawyer.



We are in process of improving the source code documentation. Code should be documented using Doxygen syntax. Some examples exist in *main.h* and *main.cpp*. In order for documentation within the code to appear in the rendered docs, the definition of the object must be in a header file, although the documentation itself can (and should) be in the source file.

## 22.1 Example Documentation Blocks

**file.h**

```
// functions
int main(int argc, char *argv[]);
```

**file.cpp** (with markdown)

```
/**
 * @brief Main application entry point.
 * @param argc The number of arguments.
 * @param argv The arguments.
 *
 * EXAMPLES:
 * ```cpp
 * main(1, const char* args[] = {"hello", "markdown", nullptr});
 * ```
 */
int main(int argc, char *argv[]) {
    // do stuff
}
```

**file.cpp** (with ReStructuredText)

```
/**
 * @brief Main application entry point.
 * @param argc The number of arguments.
 * @param argv The arguments.
 * @rst
 * EXAMPLES:
 *
 * .. code-block:: cpp
 *     main(1, const char* args[] = {"hello", "rst", nullptr});
```

(continues on next page)

(continued from previous page)

```
* @endrst
*/
int main(int argc, char *argv[]) {
    // do stuff
}
```

## 22.2 Code

### 22.2.1 main

#### Defines

**MAIL**(x)

#### Functions

void **log\_flush**()

Flush the log.

EXAMPLES:

```
log_flush();
```

int **main**(int argc, char \*argv[])

Main application entry point.

EXAMPLES:

```
main(1, const char* args[] = {"sunshine", nullptr});
```

#### Parameters

- **argc** – The number of arguments.
- **argv** – The arguments.

std::uint16\_t **map\_port**(int port)

Map a specified port based on the base port.

EXAMPLES:

```
std::uint16_t mapped_port = map_port(1);
```

#### Parameters

**port** – The port to map as a difference from the base port.

#### Returns

std::uint16\_t : The mapped port number.

void **open\_url**(const std::string &url)

void **print\_help**(const char \*name)

Print help to stdout.

EXAMPLES:

```
print_help("sunshine");
```

#### Parameters

**name** – The name of the program.

std::string **read\_file**(const char \*path)

Read a file to string.

EXAMPLES:

```
std::string contents = read_file("path/to/file");
```

#### Parameters

**path** – The path of the file.

#### Returns

std::string : The contents of the file.

void **tray\_donate\_github\_cb**(struct tray\_menu \*item)

void **tray\_donate\_mee6\_cb**(struct tray\_menu \*item)

void **tray\_donate\_patreon\_cb**(struct tray\_menu \*item)

void **tray\_donate\_paypal\_cb**(struct tray\_menu \*item)

void **tray\_open\_ui\_cb**(struct tray\_menu \*item)

void **tray\_quit\_cb**(struct tray\_menu \*item)

int **write\_file**(const char \*path, const std::string\_view &contents)

Writes a file.

EXAMPLES:

```
int write_status = write_file("path/to/file", "file contents");
```

#### Parameters

- **path** – The path of the file.
- **contents** – The contents to write.

#### Returns

int : 0 on success, -1 on failure.

### Variables

boost::log::sources::severity\_logger<int> **debug**

bool **display\_cursor**

boost::log::sources::severity\_logger<int> **error**

boost::log::sources::severity\_logger<int> **fatal**

boost::log::sources::severity\_logger<int> **info**

*thread\_pool\_util::ThreadPool* **task\_pool**

boost::log::sources::severity\_logger<int> **verbose**

boost::log::sources::severity\_logger<int> **warning**

namespace **mail**

### Functions

**MAIL**(audio\_packets)

**MAIL**(broadcast\_shutdown)

**MAIL**(hdr)

**MAIL**(idr)

**MAIL**(rumble)

**MAIL**(shutdown)

**MAIL**(switch\_display)

**MAIL**(touch\_port)

**MAIL**(video\_packets)

### Variables

*safe::mail\_t* **man**



## 22.2.2 audio

namespace **audio**

### Typedefs

using **buffer\_t** = util::buffer\_t<std::uint8\_t>

using **packet\_t** = std::pair<void\*, *buffer\_t*>

### Enums

enum **stream\_config\_e**

*Values:*

enumerator **STEREO**

enumerator **HIGH\_STEREO**

enumerator **SURROUND51**

enumerator **HIGH\_SURROUND51**

enumerator **SURROUND71**

enumerator **HIGH\_SURROUND71**

enumerator **MAX\_STREAM\_CONFIG**

struct **config\_t**

### Public Types

enum **flags\_e**

*Values:*

enumerator **HIGH\_QUALITY**

enumerator **HOST\_AUDIO**

enumerator **MAX\_FLAGS**

**Public Members**

int **channels**

std::bitset<*MAX\_FLAGS*> **flags**

int **mask**

int **packetDuration**

struct **opus\_stream\_config\_t**

**Public Members**

int **bitrate**

int **channelCount**

int **coupledStreams**

const std::uint8\_t \***mapping**

std::int32\_t **sampleRate**

int **streams**

### 22.2.3 cbs

namespace **cbs**

struct **h264\_t**

**Public Members**

*nal\_t* **sps**

struct **hevc\_t**

**Public Members***nal\_t* **sps***nal\_t* **vps**struct **nal\_t****Public Members**util::buffer\_t<std::uint8\_t> **\_new**util::buffer\_t<std::uint8\_t> **old****22.2.4 config**namespace **config**struct **audio\_t****Public Members**std::string **sink**std::string **virtual\_sink**struct **input\_t****Public Members**std::chrono::milliseconds **back\_button\_timeout**bool **controller**std::string **gamepad**std::chrono::milliseconds **key\_repeat\_delay**std::chrono::duration<double> **key\_repeat\_period**

std::unordered\_map<int, int> **keybindings**

bool **keyboard**

bool **mouse**

struct **nvhttp\_t**

### Public Members

std::string **cert**

std::string **external\_ip**

std::string **file\_state**

std::vector<int> **fps**

std::string **origin\_pin\_allowed**

std::string **origin\_web\_ui\_allowed**

std::string **pkey**

std::vector<std::string> **resolutions**

std::string **sunshine\_name**

struct **prep\_cmd\_t**

### Public Functions

inline explicit **prep\_cmd\_t**(std::string &&do\_cmd)

inline **prep\_cmd\_t**(std::string &&do\_cmd, std::string &&undo\_cmd)

### Public Members

std::string **do\_cmd**

std::string **undo\_cmd**

struct **stream\_t**

### Public Members

int **channels**

int **fec\_percentage**

std::string **file\_apps**

std::chrono::milliseconds **ping\_timeout**

struct **sunshine\_t**

### Public Members

struct *config::sunshine\_t::cmd\_t* **cmd**

std::string **config\_file**

std::string **credentials\_file**

std::bitset<*flag::FLAG\_SIZE*> **flags**

std::string **log\_file**

int **min\_log\_level**

std::string **password**

std::uint16\_t **port**

std::vector<*prep\_cmd\_t*> **prep\_cmds**

std::string **salt**

std::string **username**

struct **cmd\_t**

### Public Members

int **argc**

char \*\***argv**

std::string **name**

struct **video\_t**

### Public Members

std::string **adapter\_name**

struct *config::video\_t*::[anonymous] **amd**

int **amd\_coder**

std::optional<int> **amd\_prealanalysis**

std::optional<int> **amd\_quality\_h264**

std::optional<int> **amd\_quality\_hevc**

std::optional<int> **amd\_rc\_h264**

std::optional<int> **amd\_rc\_hevc**

std::optional<int> **amd\_usage\_h264**

std::optional<int> **amd\_usage\_hevc**

std::optional<int> **amd\_vbaq**

std::string **capture**

bool **dwmflush**

```
std::string encoder

int hevc_mode

int min_threads

struct config::video_t::[anonymous] nv

int nv_coder

std::optional<int> nv_preset

std::optional<int> nv_rc

std::optional<int> nv_tune

std::string output_name

int qp

struct config::video_t::[anonymous] qsv

std::optional<int> qsv_cavlc

std::optional<int> qsv_preset

struct config::video_t::[anonymous] sw

std::string sw_preset

std::string sw_tune

struct config::video_t::[anonymous] vt

int vt_allow_sw

int vt_coder

int vt_realtime

int vt_require_sw

namespace flag
```

## Enums

enum **flag\_e**

*Values:*

enumerator **PIN\_STDIN**

enumerator **FRESH\_STATE**

enumerator **FORCE\_VIDEO\_HEADER\_REPLACE**

enumerator **UPNP**

enumerator **CONST\_PIN**

enumerator **FLAG\_SIZE**

## 22.2.5 confighttp

### Defines

**WEB\_DIR**

### Variables

```
const std::map<std::string, std::string> mime_types = { {"css", "text/css"}, {"gif", "image/gif"}, {"htm",  
"text/html"}, {"html", "text/html"}, {"ico", "image/x-icon"}, {"jpeg", "image/jpeg"}, {"jpg", "image/jpeg"}, {"js",  
"application/javascript"}, {"json", "application/json"}, {"png", "image/png"}, {"svg", "image/svg+xml"}, {"ttf",  
"font/ttf"}, {"txt", "text/plain"}, {"woff2", "font/woff2"}, {"xml", "text/xml"}, }
```

namespace **confighttp**

### Variables

constexpr auto **PORT\_HTTPS** = 1



## 22.2.6 crypto

namespace **crypto**

### Typedefs

using **aes\_t** = std::array<std::uint8\_t, 16>

using **bignum\_t** = util::safe\_ptr<BIGNUM, BN\_free>

using **bio\_t** = util::safe\_ptr<BIO, BIO\_free\_all>

using **cipher\_ctx\_t** = util::safe\_ptr<EVP\_CIPHER\_CTX, EVP\_CIPHER\_CTX\_free>

using **md\_ctx\_t** = util::safe\_ptr<EVP\_MD\_CTX, md\_ctx\_destroy>

using **pkey\_ctx\_t** = util::safe\_ptr<EVP\_PKEY\_CTX, EVP\_PKEY\_CTX\_free>

using **pkey\_t** = util::safe\_ptr<EVP\_PKEY, EVP\_PKEY\_free>

using **sha256\_t** = std::array<std::uint8\_t, SHA256\_DIGEST\_LENGTH>

using **x509\_store\_ctx\_t** = util::safe\_ptr<X509\_STORE\_CTX, X509\_STORE\_CTX\_free>

using **x509\_store\_t** = util::safe\_ptr<X509\_STORE, X509\_STORE\_free>

using **x509\_t** = util::safe\_ptr<X509, X509\_free>

### Variables

constexpr std::size\_t **digest\_size** = 256

class **cert\_chain\_t**

### Public Functions

void **add**(*x509\_t* &&cert)

const char \***verify**(*x509\_t::element\_type* \*cert)

### Private Members

*x509\_store\_ctx\_t* **\_cert\_ctx**

std::vector<std::pair<*x509\_t*, *x509\_store\_t*>> **\_certs**

struct **creds\_t**

### Public Members

std::string **pkey**

std::string **x509**

namespace **cipher**

### Functions

constexpr std::size\_t **round\_to\_pkcs7\_padded**(std::size\_t size)

### Variables

constexpr std::size\_t **tag\_size** = 16

class **cbc\_t** : public *crypto::cipher::cipher\_t*

### Public Functions

**cbc\_t**() = default

**cbc\_t**(*cbc\_t*&&) noexcept = default

**cbc\_t**(const *crypto::aes\_t* &key, bool padding = true)

int **encrypt**(const std::string\_view &plaintext, std::uint8\_t \*cipher, *aes\_t* \*iv)

length of cipher must be at least: round\_to\_pkcs7\_padded(plaintext.size())

return -1 on error return bytes written on success

*cbc\_t* &**operator**=(*cbc\_t*&&) noexcept = default

class **cipher\_t**

Subclassed by *crypto::cipher::cbc\_t*, *crypto::cipher::ecb\_t*, *crypto::cipher::gcm\_t*

## Public Members

*cipher\_ctx\_t* **decrypt\_ctx**

*cipher\_ctx\_t* **encrypt\_ctx**

*aes\_t* **key**

bool **padding**

class **ecb\_t** : public *crypto::cipher::cipher\_t*

## Public Functions

int **decrypt**(const std::string\_view &cipher, std::vector<std::uint8\_t> &plaintext)

**ecb\_t**() = default

**ecb\_t**(const *aes\_t* &key, bool padding = true)

**ecb\_t**(*ecb\_t*&&) noexcept = default

int **encrypt**(const std::string\_view &plaintext, std::vector<std::uint8\_t> &cipher)

*ecb\_t* &**operator**=(*ecb\_t*&&) noexcept = default

class **gcm\_t** : public *crypto::cipher::cipher\_t*

## Public Functions

int **decrypt**(const std::string\_view &cipher, std::vector<std::uint8\_t> &plaintext, *aes\_t* \*iv)

int **encrypt**(const std::string\_view &plaintext, std::uint8\_t \*tagged\_cipher, *aes\_t* \*iv)

length of cipher must be at least: round\_to\_pkcs7\_padded(plaintext.size()) +  
crypto::cipher::tag\_size

return -1 on error return bytes written on success

**gcm\_t**() = default

**gcm\_t**(const *crypto::aes\_t* &key, bool padding = true)

**gcm\_t**(*gcm\_t*&&) noexcept = default

*gcm\_t* &**operator**=(*gcm\_t*&&) noexcept = default

## 22.2.7 httpcommon

namespace **http**

## 22.2.8 input

namespace **input**

### Functions

std::shared\_ptr<input\_t> **alloc**(*safe::mail\_t* mail)

inline std::unique\_ptr<*platf::deinit\_t*> **init**()

void **passthrough**(std::shared\_ptr<input\_t> &input, std::vector<std::uint8\_t> &&input\_data)

void **print**(void \*input)

void **reset**(std::shared\_ptr<input\_t> &input)

struct **touch\_port\_t** : public *platf::touch\_port\_t*

### Public Members

float **client\_offsetX**

float **client\_offsetY**

int **env\_height**

int **env\_width**

float **scalar\_inv**

## 22.2.9 move\_by\_copy

namespace **move\_by\_copy\_util**

## Functions

```
template<class T>
MoveByCopy<T> cmove(T &movable)

template<class T>
MoveByCopy<T> const_cmove(const T &movable)

template<class T>
class MoveByCopy
```

## Public Types

```
typedef T move_type
```

## Public Functions

```
inline MoveByCopy(const MoveByCopy &other)

inline explicit MoveByCopy(move_type &&to_move)

MoveByCopy(MoveByCopy &&other) = default

inline operator move_type()

inline MoveByCopy &operator=(const MoveByCopy &other)

MoveByCopy &operator=(MoveByCopy &&other) = default
```

## Private Members

```
move_type _to_move
```

## 22.2.10 network

```
namespace net
```

## Typedefs

```
using host_t = util::safe_ptr<ENetHost, free_host>

using packet_t = util::safe_ptr<ENetPacket, enet_packet_destroy>

using peer_t = ENetPeer*
```

## Enums

enum **net\_e**

*Values:*

enumerator **PC**

enumerator **LAN**

enumerator **WAN**

### 22.2.11 nvhttp

namespace **nvhttp**

This namespace contains all the functions and variables related to the nvhttp (GameStream) server.

#### Variables

constexpr auto **GFE\_VERSION** = "3.23.0.74"

The GFE version we are replicating.

constexpr auto **PORT\_HTTP** = 0

The HTTP port, as a difference from the config port.

constexpr auto **PORT\_HTTPS** = -5

The HTTPS port, as a difference from the config port.

constexpr auto **VERSION** = "7.1.431.-1"

The protocol version.

### 22.2.12 process

#### Defines

**\_\_kernel\_entry**

namespace **proc**

## Typedefs

```
typedef config::prep_cmd_t cmd_t
```

```
using file_t = util::safe_ptr_v2<FILE, int, fclose>
```

```
struct ctx_t
```

## Public Members

```
std::string cmd
```

```
std::vector<std::string> detached
```

Some applications, such as Steam, either exit quickly, or keep running indefinitely. Steam.exe is one such application. That is why some applications need be run and forgotten about

```
std::string id
```

```
std::string image_path
```

```
std::string name
```

```
std::string output
```

```
std::vector<cmd_t> prep_cmds
```

```
std::string working_dir
```

```
class proc_t
```

## Public Functions

```
int execute(int app_id)
```

```
std::string get_app_image(int app_id)
```

```
std::vector<ctx_t> &get_apps()
```

```
const std::vector<ctx_t> &get_apps() const
```

```
inline proc_t(boost::process::environment &&env, std::vector<ctx_t> &&apps)
```

```
int running()
```

### Returns

\_app\_id if a process is running, otherwise returns 0

```
void terminate()
```

```
~proc_t()
```

### Private Members

```
ctx_t _app
```

```
int _app_id
```

```
std::vector<cmd_t>::const_iterator _app_prep_begin
```

```
std::vector<cmd_t>::const_iterator _app_prep_it
```

```
std::vector<ctx_t> _apps
```

```
boost::process::environment _env
```

```
file_t _pipe
```

```
boost::process::child _process
```

```
boost::process::group _process_handle
```

```
bool placebo = { }
```

## 22.2.13 round\_robin

```
namespace round_robin_util
```

### Functions

```
template<class V, class It>  
round_robin_t<V, It> make_round_robin(It begin, It end)
```

```
template<class V, class T>
```

```
class it_wrap_t
```



## Public Types

```
typedef std::ptrdiff_t diff_t

using difference_type = V

typedef T iterator

using iterator_category = std::random_access_iterator_tag

using pointer = V*

using reference = V&

using value_type = V
```

## Public Functions

```
inline bool operator!=(const iterator &other) const

inline reference operator*()

inline const reference operator*() const

inline iterator operator+(diff_t step)

inline iterator operator++()

inline iterator operator++(int)

inline iterator operator+=(diff_t step)

inline iterator operator-(diff_t step)

inline diff_t operator-(iterator first)

inline iterator operator--()

inline iterator operator--(int)

inline iterator operator-=(diff_t step)

inline pointer operator->()

inline const pointer operator->() const

inline bool operator<(const iterator &other) const

inline bool operator<=(const iterator &other) const

inline bool operator==(const iterator &other) const

inline bool operator>(const iterator &other) const

inline bool operator>=(const iterator &other) const
```

### Private Functions

```
inline iterator &_this()

inline const iterator &_this() const

template<class V, class It>

class round_robin_t : public round_robin_util::it_wrap_t<V, round_robin_t<V, It>>
```

### Public Types

```
using iterator = It

using pointer = V*
```

### Public Functions

```
inline void dec()

inline bool eq(const round_robin_t &other) const

inline pointer get() const

inline void inc()

inline round_robin_t(iterator begin, iterator end)
```

### Private Members

```
It _begin

It _end

It _pos
```

## 22.2.14 rtsp

```
namespace rtsp_stream
```

### Variables

```
constexpr auto RTSP_SETUP_PORT = 21
```

```
struct launch_session_t
```

### Public Members

```
crypto::aes_t gcm_key
```

```
bool host_audio
```

```
crypto::aes_t iv
```

## 22.2.15 stream

```
namespace stream
```

### Variables

```
constexpr auto AUDIO_STREAM_PORT = 11
```

```
constexpr auto CONTROL_PORT = 10
```

```
constexpr auto VIDEO_STREAM_PORT = 9
```

```
struct config_t
```

### Public Members

```
audio::config_t audio
```

```
int audioQosType
```

```
int controlProtocolType
```

```
int featureFlags
```

```
std::optional<int> gcmmap
```

```
int minRequiredFecPackets
```

```
video::config_t monitor
```

```
int packetSize
```

```
int videoQosType
```

```
namespace session
```

### Enums

```
enum class state_e : int
```

```
Values:
```

```
enumerator STOPPED
```

```
enumerator STOPPING
```

```
enumerator STARTING
```

```
enumerator RUNNING
```

## 22.2.16 sync

```
namespace sync_util
```

```
template<class T, class M = std::mutex>
```

```
class sync_t
```

### Public Types

```
using mutex_t = M
```

```
using value_t = T
```

**Public Functions**

```

inline std::lock_guard<mutex_t> lock()

inline value_t &operator*()

inline const value_t &operator*() const

inline value_t *operator->()

inline sync_t &operator=(const value_t &val) noexcept

inline sync_t &operator=(sync_t &&other) noexcept

inline sync_t &operator=(sync_t &other) noexcept

template<class V>
inline sync_t &operator=(V &&val)

inline sync_t &operator=(value_t &&val) noexcept

template<class ...Args>
inline sync_t(Args&&... args)

```

**Public Members**

```
value_t raw
```

**Private Members**

```
mutex_t _lock
```

**22.2.17 system\_tray****22.2.18 tasl\_pool**

```
namespace task_pool_util
```

```
template<class Function>
```

```
class _Impl : public task_pool_util::_ImplBase
```

### Public Functions

inline **\_Impl**(*Function* &&f)

inline virtual void **run**() override

### Private Members

*Function* **\_func**

class **\_ImplBase**

Subclassed by *task\_pool\_util::Impl< Function >*

### Public Functions

virtual void **run**() = 0

inline virtual **~\_ImplBase**() = default

class **TaskPool**

Subclassed by *thread\_pool\_util::ThreadPool*

### Public Types

typedef std::unique\_ptr<*\_ImplBase*> **\_\_task**

typedef std::chrono::steady\_clock::time\_point **\_\_time\_point**

typedef *\_ImplBase* \***task\_id\_t**

### Public Functions

inline bool **cancel**(*task\_id\_t* task\_id)

template<class **X**, class **Y**>

inline void **delay**(*task\_id\_t* task\_id, std::chrono::duration<*X*, *Y*> duration)

#### Parameters

**duration** – The delay before executing the task

inline std::optional<*\_\_time\_point*> **next**()

inline *TaskPool* &**operator**=(*TaskPool* &&other) noexcept

inline std::optional<*\_\_task*> **pop**()

inline std::optional<std::pair<*\_\_time\_point*, *\_\_task*>> **pop**(*task\_id\_t* task\_id)

template<class **Function**, class ...**Args**>

```

inline auto push(Function &&newTask, Args&&... args)

template<class Function, class X, class Y, class ...Args>
inline auto pushDelayed(Function &&newTask, std::chrono::duration<X, Y> duration, Args&&... args)
    Returns
        an id to potentially delay the task

inline void pushDelayed(std::pair<__time_point, __task> &&task)

inline bool ready()

TaskPool() = default

inline TaskPool(TaskPool &&other) noexcept

```

### Protected Attributes

```

std::mutex _task_mutex

std::deque<__task> _tasks

std::vector<std::pair<__time_point, __task>> _timer_tasks

```

### Private Functions

```

template<class Function>
inline std::unique_ptr<__ImplBase> toRunnable(Function &&f)

template<class R>
class timer_task_t

```

### Public Functions

```

inline timer_task_t(task_id_t task_id, std::future<R> &future)

```

### Public Members

```

std::future<R> future

task_id_t task_id

```

## 22.2.19 thread\_pool

namespace **thread\_pool\_util**

class **ThreadPool** : public *task\_pool\_util::TaskPool*

### Public Types

typedef TaskPool::\_\_task \_\_**task**

### Public Functions

inline void **\_main**()

inline void **join**()

template<class **Function**, class ...**Args**>

inline auto **push**(*Function* &&newTask, *Args*&&... args)

template<class **Function**, class **X**, class **Y**, class ...**Args**>

inline auto **pushDelayed**(*Function* &&newTask, std::chrono::duration<*X*, *Y*> duration, *Args*&&... args)

inline void **pushDelayed**(std::pair<\_\_time\_point, \_\_*task*> &&task)

inline void **start**(int threads)

inline void **stop**()

inline **ThreadPool**()

inline explicit **ThreadPool**(int threads)

inline ~**ThreadPool**() noexcept

### Private Members

bool **\_continue**

std::condition\_variable **\_cv**

std::mutex **\_lock**

std::vector<std::thread> **\_thread**



## 22.2.20 thread\_safe

namespace **safe**

### Typedefs

```
template<class T>
using alarm_t = std::shared_ptr<alarm_raw_t<T>>

using mail_t = std::shared_ptr<mail_raw_t>

using signal_t = event_t<bool>
```

### Functions

```
inline void cleanup(mail_raw_t*)

template<class T>
inline auto lock(const std::weak_ptr<void> &wp)

template<class T>
alarm_t<T> make_alarm()

template<class T, class F_Construct, class F_Destruct>
auto make_shared(F_Construct &&fc, F_Destruct &&fd)

template<class T>
class alarm_raw_t
```

### Public Types

```
using status_t = util::optional_t<T>
```

### Public Functions

```
inline alarm_raw_t()

inline void reset()

inline void ring(const status_t &status)

inline void ring(status_t &&status)

inline status_t &status()

inline const status_t &status() const
```

```
inline auto wait()

template<class Pred>
inline auto wait(Pred &&pred)

template<class Rep, class Period>
inline auto wait_for(const std::chrono::duration<Rep, Period> &rel_time)

template<class Rep, class Period, class Pred>
inline auto wait_for(const std::chrono::duration<Rep, Period> &rel_time, Pred &&pred)

template<class Rep, class Period>
inline auto wait_until(const std::chrono::duration<Rep, Period> &rel_time)

template<class Rep, class Period, class Pred>
inline auto wait_until(const std::chrono::duration<Rep, Period> &rel_time, Pred &&pred)
```

## Private Members

```
std::condition_variable _cv
```

```
std::mutex _lock
```

```
status_t _status
```

```
template<class T>
```

```
class event_t
```

## Public Types

```
using status_t = util::optional_t<T>
```

## Public Functions

```
inline bool peek()
```

```
inline status_t pop()
```

```
template<class Rep, class Period>
inline status_t pop(std::chrono::duration<Rep, Period> delay)
```

```
template<class ...Args>
inline void raise(Args&&... args)
```

```
inline void reset()
```

```
inline bool running() const
```

```
inline void stop()
```

```
inline const status_t &view()
```

### Private Members

```
bool _continue = {true}
```

```
std::condition_variable _cv
```

```
std::mutex _lock
```

```
status_t _status = {util::false_v<status_t>}
```

```
class mail_raw_t : public std::enable_shared_from_this<mail_raw_t>
```

### Public Types

```
template<class T>
```

```
using event_t = std::shared_ptr<post_t<event_t<T>>>
```

```
template<class T>
```

```
using queue_t = std::shared_ptr<post_t<queue_t<T>>>
```

### Public Functions

```
inline void cleanup()
```

```
template<class T>
```

```
inline event_t<T> event(const std::string_view &id)
```

```
template<class T>
```

```
inline queue_t<T> queue(const std::string_view &id)
```

### Public Members

```
std::map<std::string, std::weak_ptr<void>, std::less<>> id_to_post
```

```
std::mutex mutex
```

```
template<class T>
```

```
class post_t : public T
```

### Public Functions

```
template<class ...Args>
inline post_t(mail_t mail, Args&&... args)

inline ~post_t()
```

### Public Members

```
mail_t mail
```

```
template<class T>
class queue_t
```

### Public Types

```
using status_t = util::optional_t<T>
```

### Public Functions

```
inline bool peek()

inline status_t pop()

template<class Rep, class Period>
inline status_t pop(std::chrono::duration<Rep, Period> delay)

inline queue_t(std::uint32_t max_elements = 32)

template<class ...Args>
inline void raise(Args&&... args)

inline bool running() const

inline void stop()

inline std::vector<T> &unsafe()
```

### Private Members

```
bool _continue = {true}

std::condition_variable _cv

std::mutex _lock

std::uint32_t _max_elements
```

```

std::vector<T> _queue

template<class T>

class shared_t

```

### Public Types

```

using construct_f = std::function<int(element_type&)>

using destruct_f = std::function<void(element_type&)>

using element_type = T

```

### Public Functions

```

inline ptr_t ref()

template<class FC, class FD>
inline shared_t(FC &&fc, FD &&fd)

```

### Private Members

```

construct_f _construct

std::uint32_t _count

destruct_f _destruct

std::mutex _lock

std::array<std::uint8_t, sizeof(element_type)> _object_buf

struct ptr_t

```

### Public Functions

```

inline element_type *get() const

inline operator bool() const

inline element_type *operator->()

inline ptr_t &operator=(const ptr_t &ptr) noexcept

```

```
inline ptr_t &operator=(ptr_t &&ptr) noexcept
inline ptr_t()
inline ptr_t(const ptr_t &ptr) noexcept
inline ptr_t(ptr_t &&ptr) noexcept
inline explicit ptr_t(shared_t *owner)
inline void release()
inline ~ptr_t()
```

### Public Members

```
shared_t *owner
```

## 22.2.21 upnp

```
namespace upnp
```

## 22.2.22 utility

---

**Todo:** Add utility.h

---

## 22.2.23 uuid

```
namespace uuid_util
```

```
union uuid_t
```

### Public Functions

```
inline constexpr bool operator<(const uuid_t &other) const
inline constexpr bool operator==(const uuid_t &other) const
inline constexpr bool operator>(const uuid_t &other) const
inline std::string string() const
```

**Public Members**

std::uint16\_t **b16**[8]

std::uint32\_t **b32**[4]

std::uint64\_t **b64**[2]

std::uint8\_t **b8**[16]

**Public Static Functions**

static inline *uuid\_t* **generate**()

static inline *uuid\_t* **generate**(std::default\_random\_engine &engine)

**22.2.24 video**

namespace **video**

**Typedefs**

using **float2** = float[2]

using **float3** = float[3]

using **float4** = float[4]

using **hdr\_info\_t** = std::unique\_ptr<*hdr\_info\_raw\_t*>

using **packet\_t** = std::unique\_ptr<*packet\_raw\_t*>

struct **color\_t**

**Public Members**

*float4* **color\_vec\_u**

*float4* **color\_vec\_v**

*float4* **color\_vec\_y**

*float2* **range\_uv**

*float2* **range\_y**

struct **config\_t**

### **Public Members**

int **bitrate**

int **dynamicRange**

int **encoderCscMode**

int **framerate**

int **height**

int **numRefFrames**

int **slicesPerFrame**

int **videoFormat**

int **width**

struct **hdr\_info\_raw\_t**

### **Public Functions**

inline explicit **hdr\_info\_raw\_t**(bool enabled)

inline explicit **hdr\_info\_raw\_t**(bool enabled, const SS\_HDR\_METADATA &metadata)

### **Public Members**

bool **enabled**

SS\_HDR\_METADATA **metadata**

struct **packet\_raw\_t**



**Public Functions**

```

inline void init_packet()

template<class P>
inline explicit packet_raw_t(P *user_data)

inline explicit packet_raw_t(std::nullptr_t)

inline ~packet_raw_t()

```

**Public Members**

```

AVPacket *av_packet

void *channel_data

std::vector<replace_t> *replacements

struct replace_t

```

**Public Functions**

```

inline replace_t(std::string_view old, std::string_view _new) noexcept

```

**Public Members**

```

std::string_view _new

std::string_view old

```

**22.2.25 platform**

**common**

---

**Todo:** Add common.h

---

linux

cuda

graphics

---

**Todo:** Add graphics.h

---

misc

---

**Todo:** Add misc.h

---

vaapi

namespace **egl**

namespace **va**

wayland

namespace **wl**

class **monitor\_t**

### Public Functions

void **listen**(zxdg\_output\_manager\_v1 \*output\_manager)

**monitor\_t**(const *monitor\_t*&) = delete

**monitor\_t**(*monitor\_t*&&) = delete

inline **monitor\_t**(wl\_output \*output)

*monitor\_t* &**operator**=(const *monitor\_t*&) = delete

*monitor\_t* &**operator**=(*monitor\_t*&&) = delete

## Public Members

std::string **description**

std::string **name**

wl\_output \***output**

*platf::touch\_port\_t* **viewport**

## x11grab

---

**Todo:** Add x11grab.h

---

## macos

### av\_audio

---

**Todo:** Add av\_audio.h

---

### av\_img\_t

---

**Todo:** Add av\_img\_t.h

---

### av\_video

---

**Todo:** Add av\_video.h

---

## misc

namespace **dyn**

## Sunshine

---

### nv12\_zero\_device

---

**Todo:** Add nv12\_zero\_device.h

---

### windows

#### display

---

**Todo:** Add display.h

---

### misc

namespace **platf**

### PolicyConfig

---

**Todo:** Add PolicyConfig.h

---

## Symbols

`__kernel_entry` (*C macro*), 112

## A

`audio` (*C++ type*), 99  
`audio::buffer_t` (*C++ type*), 99  
`audio::config_t` (*C++ struct*), 99  
`audio::config_t::channels` (*C++ member*), 100  
`audio::config_t::flags` (*C++ member*), 100  
`audio::config_t::flags_e` (*C++ enum*), 99  
`audio::config_t::flags_e::HIGH_QUALITY` (*C++ enumerator*), 99  
`audio::config_t::flags_e::HOST_AUDIO` (*C++ enumerator*), 99  
`audio::config_t::flags_e::MAX_FLAGS` (*C++ enumerator*), 99  
`audio::config_t::mask` (*C++ member*), 100  
`audio::config_t::packetDuration` (*C++ member*), 100  
`audio::opus_stream_config_t` (*C++ struct*), 100  
`audio::opus_stream_config_t::bitrate` (*C++ member*), 100  
`audio::opus_stream_config_t::channelCount` (*C++ member*), 100  
`audio::opus_stream_config_t::coupledStreams` (*C++ member*), 100  
`audio::opus_stream_config_t::mapping` (*C++ member*), 100  
`audio::opus_stream_config_t::sampleRate` (*C++ member*), 100  
`audio::opus_stream_config_t::streams` (*C++ member*), 100  
`audio::packet_t` (*C++ type*), 99  
`audio::stream_config_e` (*C++ enum*), 99  
`audio::stream_config_e::HIGH_STEREO` (*C++ enumerator*), 99  
`audio::stream_config_e::HIGH_SURROUND51` (*C++ enumerator*), 99  
`audio::stream_config_e::HIGH_SURROUND71` (*C++ enumerator*), 99  
`audio::stream_config_e::MAX_STREAM_CONFIG` (*C++ enumerator*), 99

`audio::stream_config_e::STEREO` (*C++ enumerator*), 99  
`audio::stream_config_e::SURROUND51` (*C++ enumerator*), 99  
`audio::stream_config_e::SURROUND71` (*C++ enumerator*), 99

## C

`cbs` (*C++ type*), 100  
`cbs::h264_t` (*C++ struct*), 100  
`cbs::h264_t::sps` (*C++ member*), 100  
`cbs::hevc_t` (*C++ struct*), 100  
`cbs::hevc_t::sps` (*C++ member*), 101  
`cbs::hevc_t::vps` (*C++ member*), 101  
`cbs::nal_t` (*C++ struct*), 101  
`cbs::nal_t::_new` (*C++ member*), 101  
`cbs::nal_t::old` (*C++ member*), 101  
`config` (*C++ type*), 101  
`config::audio_t` (*C++ struct*), 101  
`config::audio_t::sink` (*C++ member*), 101  
`config::audio_t::virtual_sink` (*C++ member*), 101  
`config::flag` (*C++ type*), 105  
`config::flag::flag_e` (*C++ enum*), 106  
`config::flag::flag_e::CONST_PIN` (*C++ enumerator*), 106  
`config::flag::flag_e::FLAG_SIZE` (*C++ enumerator*), 106  
`config::flag::flag_e::FORCE_VIDEO_HEADER_REPLACE` (*C++ enumerator*), 106  
`config::flag::flag_e::FRESH_STATE` (*C++ enumerator*), 106  
`config::flag::flag_e::PIN_STDIN` (*C++ enumerator*), 106  
`config::flag::flag_e::UPNP` (*C++ enumerator*), 106  
`config::input_t` (*C++ struct*), 101  
`config::input_t::back_button_timeout` (*C++ member*), 101  
`config::input_t::controller` (*C++ member*), 101  
`config::input_t::gamepad` (*C++ member*), 101

`config::input_t::key_repeat_delay` (C++ member), 101  
`config::input_t::key_repeat_period` (C++ member), 101  
`config::input_t::keybindings` (C++ member), 101  
`config::input_t::keyboard` (C++ member), 102  
`config::input_t::mouse` (C++ member), 102  
`config::nvhttp_t` (C++ struct), 102  
`config::nvhttp_t::cert` (C++ member), 102  
`config::nvhttp_t::external_ip` (C++ member), 102  
`config::nvhttp_t::file_state` (C++ member), 102  
`config::nvhttp_t::fps` (C++ member), 102  
`config::nvhttp_t::origin_pin_allowed` (C++ member), 102  
`config::nvhttp_t::origin_web_ui_allowed` (C++ member), 102  
`config::nvhttp_t::pkey` (C++ member), 102  
`config::nvhttp_t::resolutions` (C++ member), 102  
`config::nvhttp_t::sunshine_name` (C++ member), 102  
`config::prep_cmd_t` (C++ struct), 102  
`config::prep_cmd_t::do_cmd` (C++ member), 103  
`config::prep_cmd_t::prep_cmd_t` (C++ function), 102  
`config::prep_cmd_t::undo_cmd` (C++ member), 103  
`config::stream_t` (C++ struct), 103  
`config::stream_t::channels` (C++ member), 103  
`config::stream_t::fec_percentage` (C++ member), 103  
`config::stream_t::file_apps` (C++ member), 103  
`config::stream_t::ping_timeout` (C++ member), 103  
`config::sunshine_t` (C++ struct), 103  
`config::sunshine_t::cmd` (C++ member), 103  
`config::sunshine_t::cmd_t` (C++ struct), 104  
`config::sunshine_t::cmd_t::argc` (C++ member), 104  
`config::sunshine_t::cmd_t::argv` (C++ member), 104  
`config::sunshine_t::cmd_t::name` (C++ member), 104  
`config::sunshine_t::config_file` (C++ member), 103  
`config::sunshine_t::credentials_file` (C++ member), 103  
`config::sunshine_t::flags` (C++ member), 103  
`config::sunshine_t::log_file` (C++ member), 103  
`config::sunshine_t::min_log_level` (C++ member), 103  
`config::sunshine_t::password` (C++ member), 103  
`config::sunshine_t::port` (C++ member), 103  
`config::sunshine_t::prep_cmds` (C++ member), 103  
`config::sunshine_t::salt` (C++ member), 103  
`config::sunshine_t::username` (C++ member), 103  
`config::video_t` (C++ struct), 104  
`config::video_t::adapter_name` (C++ member), 104  
`config::video_t::amd` (C++ member), 104  
`config::video_t::amd_coder` (C++ member), 104  
`config::video_t::amd_prealanalysis` (C++ member), 104  
`config::video_t::amd_quality_h264` (C++ member), 104  
`config::video_t::amd_quality_hevc` (C++ member), 104  
`config::video_t::amd_rc_h264` (C++ member), 104  
`config::video_t::amd_rc_hevc` (C++ member), 104  
`config::video_t::amd_usage_h264` (C++ member), 104  
`config::video_t::amd_usage_hevc` (C++ member), 104  
`config::video_t::amd_vbaq` (C++ member), 104  
`config::video_t::capture` (C++ member), 104  
`config::video_t::dwmflush` (C++ member), 104  
`config::video_t::encoder` (C++ member), 104  
`config::video_t::hevc_mode` (C++ member), 105  
`config::video_t::min_threads` (C++ member), 105  
`config::video_t::nv` (C++ member), 105  
`config::video_t::nv_coder` (C++ member), 105  
`config::video_t::nv_preset` (C++ member), 105  
`config::video_t::nv_rc` (C++ member), 105  
`config::video_t::nv_tune` (C++ member), 105  
`config::video_t::output_name` (C++ member), 105  
`config::video_t::qp` (C++ member), 105  
`config::video_t::qsv` (C++ member), 105  
`config::video_t::qsv_cavlc` (C++ member), 105  
`config::video_t::qsv_preset` (C++ member), 105  
`config::video_t::sw` (C++ member), 105  
`config::video_t::sw_preset` (C++ member), 105  
`config::video_t::sw_tune` (C++ member), 105  
`config::video_t::vt` (C++ member), 105  
`config::video_t::vt_allow_sw` (C++ member), 105  
`config::video_t::vt_coder` (C++ member), 105  
`config::video_t::vt_realtime` (C++ member), 105  
`config::video_t::vt_require_sw` (C++ member), 105  
`confighttp` (C++ type), 106  
`confighttp::PORT_HTTPS` (C++ member), 106  
`crypto` (C++ type), 107  
`crypto::aes_t` (C++ type), 107  
`crypto::bignum_t` (C++ type), 107  
`crypto::bio_t` (C++ type), 107  
`crypto::cert_chain_t` (C++ class), 107  
`crypto::cert_chain_t::_cert_ctx` (C++ member), 108

- crypto::cert\_chain\_t::\_certs (C++ member), 108  
 crypto::cert\_chain\_t::add (C++ function), 107  
 crypto::cert\_chain\_t::verify (C++ function), 107  
 crypto::cipher (C++ type), 108  
 crypto::cipher::cbc\_t (C++ class), 108  
 crypto::cipher::cbc\_t::cbc\_t (C++ function), 108  
 crypto::cipher::cbc\_t::encrypt (C++ function), 108  
 crypto::cipher::cbc\_t::operator= (C++ function), 108  
 crypto::cipher::cipher\_t (C++ class), 108  
 crypto::cipher::cipher\_t::decrypt\_ctx (C++ member), 109  
 crypto::cipher::cipher\_t::encrypt\_ctx (C++ member), 109  
 crypto::cipher::cipher\_t::key (C++ member), 109  
 crypto::cipher::cipher\_t::padding (C++ member), 109  
 crypto::cipher::ecb\_t (C++ class), 109  
 crypto::cipher::ecb\_t::decrypt (C++ function), 109  
 crypto::cipher::ecb\_t::ecb\_t (C++ function), 109  
 crypto::cipher::ecb\_t::encrypt (C++ function), 109  
 crypto::cipher::ecb\_t::operator= (C++ function), 109  
 crypto::cipher::gcm\_t (C++ class), 109  
 crypto::cipher::gcm\_t::decrypt (C++ function), 109  
 crypto::cipher::gcm\_t::encrypt (C++ function), 109  
 crypto::cipher::gcm\_t::gcm\_t (C++ function), 109  
 crypto::cipher::gcm\_t::operator= (C++ function), 109  
 crypto::cipher::round\_to\_pkcs7\_padded (C++ function), 108  
 crypto::cipher::tag\_size (C++ member), 108  
 crypto::cipher\_ctx\_t (C++ type), 107  
 crypto::creds\_t (C++ struct), 108  
 crypto::creds\_t::pkey (C++ member), 108  
 crypto::creds\_t::x509 (C++ member), 108  
 crypto::digest\_size (C++ member), 107  
 crypto::md\_ctx\_t (C++ type), 107  
 crypto::pkey\_ctx\_t (C++ type), 107  
 crypto::pkey\_t (C++ type), 107  
 crypto::sha256\_t (C++ type), 107  
 crypto::x509\_store\_ctx\_t (C++ type), 107  
 crypto::x509\_store\_t (C++ type), 107  
 crypto::x509\_t (C++ type), 107
- ## D
- debug (C++ member), 98  
 display\_cursor (C++ member), 98
- dyn (C++ type), 133
- ## E
- egl (C++ type), 132  
 error (C++ member), 98
- ## F
- fatal (C++ member), 98
- ## H
- http (C++ type), 110
- ## I
- info (C++ member), 98  
 input (C++ type), 110  
 input::alloc (C++ function), 110  
 input::init (C++ function), 110  
 input::passthrough (C++ function), 110  
 input::print (C++ function), 110  
 input::reset (C++ function), 110  
 input::touch\_port\_t (C++ struct), 110  
 input::touch\_port\_t::client\_offsetX (C++ member), 110  
 input::touch\_port\_t::client\_offsetY (C++ member), 110  
 input::touch\_port\_t::env\_height (C++ member), 110  
 input::touch\_port\_t::env\_width (C++ member), 110  
 input::touch\_port\_t::scalar\_inv (C++ member), 110
- ## L
- log\_flush (C++ function), 96
- ## M
- MAIL (C macro), 96  
 mail (C++ type), 98  
 mail::MAIL (C++ function), 98  
 mail::man (C++ member), 98  
 main (C++ function), 96  
 map\_port (C++ function), 96  
 mime\_types (C++ member), 106  
 move\_by\_copy\_util (C++ type), 110  
 move\_by\_copy\_util::cmove (C++ function), 111  
 move\_by\_copy\_util::const\_cmove (C++ function), 111  
 move\_by\_copy\_util::MoveByCopy (C++ class), 111  
 move\_by\_copy\_util::MoveByCopy::\_to\_move (C++ member), 111  
 move\_by\_copy\_util::MoveByCopy::move\_type (C++ type), 111

move\_by\_copy\_util::MoveByCopy::MoveByCopy  
(C++ function), 111  
 move\_by\_copy\_util::MoveByCopy::operator  
 move\_type (C++ function), 111  
 move\_by\_copy\_util::MoveByCopy::operator=  
 (C++ function), 111

## N

net (C++ type), 111  
 net::host\_t (C++ type), 111  
 net::net\_e (C++ enum), 112  
 net::net\_e::LAN (C++ enumerator), 112  
 net::net\_e::PC (C++ enumerator), 112  
 net::net\_e::WAN (C++ enumerator), 112  
 net::packet\_t (C++ type), 111  
 net::peer\_t (C++ type), 111  
 nvhttp (C++ type), 112  
 nvhttp::GFE\_VERSION (C++ member), 112  
 nvhttp::PORT\_HTTP (C++ member), 112  
 nvhttp::PORT\_HTTPS (C++ member), 112  
 nvhttp::VERSION (C++ member), 112

## O

open\_url (C++ function), 97

## P

platf (C++ type), 134  
 print\_help (C++ function), 97  
 proc (C++ type), 112  
 proc::cmd\_t (C++ type), 113  
 proc::ctx\_t (C++ struct), 113  
 proc::ctx\_t::cmd (C++ member), 113  
 proc::ctx\_t::detached (C++ member), 113  
 proc::ctx\_t::id (C++ member), 113  
 proc::ctx\_t::image\_path (C++ member), 113  
 proc::ctx\_t::name (C++ member), 113  
 proc::ctx\_t::output (C++ member), 113  
 proc::ctx\_t::prep\_cmds (C++ member), 113  
 proc::ctx\_t::working\_dir (C++ member), 113  
 proc::file\_t (C++ type), 113  
 proc::proc\_t (C++ class), 113  
 proc::proc\_t::\_app (C++ member), 114  
 proc::proc\_t::\_app\_id (C++ member), 114  
 proc::proc\_t::\_app\_prep\_begin (C++ member),  
 114  
 proc::proc\_t::\_app\_prep\_it (C++ member), 114  
 proc::proc\_t::\_apps (C++ member), 114  
 proc::proc\_t::\_env (C++ member), 114  
 proc::proc\_t::\_pipe (C++ member), 114  
 proc::proc\_t::\_process (C++ member), 114  
 proc::proc\_t::\_process\_handle (C++ member),  
 114  
 proc::proc\_t::~~proc\_t (C++ function), 114

proc::proc\_t::execute (C++ function), 113  
 proc::proc\_t::get\_app\_image (C++ function), 113  
 proc::proc\_t::get\_apps (C++ function), 113  
 proc::proc\_t::placebo (C++ member), 114  
 proc::proc\_t::proc\_t (C++ function), 113  
 proc::proc\_t::running (C++ function), 113  
 proc::proc\_t::terminate (C++ function), 113

## R

read\_file (C++ function), 97  
 round\_robin\_util (C++ type), 114  
 round\_robin\_util::it\_wrap\_t (C++ class), 114  
 round\_robin\_util::it\_wrap\_t::\_this (C++ func-  
 tion), 116  
 round\_robin\_util::it\_wrap\_t::diff\_t (C++  
 type), 115  
 round\_robin\_util::it\_wrap\_t::difference\_type  
 (C++ type), 115  
 round\_robin\_util::it\_wrap\_t::iterator (C++  
 type), 115  
 round\_robin\_util::it\_wrap\_t::iterator\_category  
 (C++ type), 115  
 round\_robin\_util::it\_wrap\_t::operator!=  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator\* (C++  
 function), 115  
 round\_robin\_util::it\_wrap\_t::operator+ (C++  
 function), 115  
 round\_robin\_util::it\_wrap\_t::operator++  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator+=  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator==  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator- (C++  
 function), 115  
 round\_robin\_util::it\_wrap\_t::operator-=  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator--  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator->  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator> (C++  
 function), 115  
 round\_robin\_util::it\_wrap\_t::operator>=  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::operator< (C++  
 function), 115  
 round\_robin\_util::it\_wrap\_t::operator<=  
 (C++ function), 115  
 round\_robin\_util::it\_wrap\_t::pointer (C++  
 type), 115  
 round\_robin\_util::it\_wrap\_t::reference (C++  
 type), 115



round\_robin\_util::it\_wrap\_t::value\_type (C++ type), 115  
 round\_robin\_util::make\_round\_robin (C++ function), 114  
 round\_robin\_util::round\_robin\_t (C++ class), 116  
 round\_robin\_util::round\_robin\_t::\_begin (C++ member), 116  
 round\_robin\_util::round\_robin\_t::\_end (C++ member), 116  
 round\_robin\_util::round\_robin\_t::\_pos (C++ member), 116  
 round\_robin\_util::round\_robin\_t::dec (C++ function), 116  
 round\_robin\_util::round\_robin\_t::eq (C++ function), 116  
 round\_robin\_util::round\_robin\_t::get (C++ function), 116  
 round\_robin\_util::round\_robin\_t::inc (C++ function), 116  
 round\_robin\_util::round\_robin\_t::iterator (C++ type), 116  
 round\_robin\_util::round\_robin\_t::pointer (C++ type), 116  
 round\_robin\_util::round\_robin\_t::round\_robin\_tsafe (C++ function), 116  
 rtsp\_stream (C++ type), 116  
 rtsp\_stream::launch\_session\_t (C++ struct), 117  
 rtsp\_stream::launch\_session\_t::gcm\_key (C++ member), 117  
 rtsp\_stream::launch\_session\_t::host\_audio (C++ member), 117  
 rtsp\_stream::launch\_session\_t::iv (C++ member), 117  
 rtsp\_stream::RTSP\_SETUP\_PORT (C++ member), 117

## S

safe (C++ type), 123  
 safe::alarm\_raw\_t (C++ class), 123  
 safe::alarm\_raw\_t::\_cv (C++ member), 124  
 safe::alarm\_raw\_t::\_lock (C++ member), 124  
 safe::alarm\_raw\_t::\_status (C++ member), 124  
 safe::alarm\_raw\_t::alarm\_raw\_t (C++ function), 123  
 safe::alarm\_raw\_t::reset (C++ function), 123  
 safe::alarm\_raw\_t::ring (C++ function), 123  
 safe::alarm\_raw\_t::status (C++ function), 123  
 safe::alarm\_raw\_t::status\_t (C++ type), 123  
 safe::alarm\_raw\_t::wait (C++ function), 123, 124  
 safe::alarm\_raw\_t::wait\_for (C++ function), 124  
 safe::alarm\_raw\_t::wait\_until (C++ function), 124  
 safe::alarm\_t (C++ type), 123  
 safe::cleanup (C++ function), 123  
 safe::event\_t (C++ class), 124  
 safe::event\_t::\_continue (C++ member), 125  
 safe::event\_t::\_cv (C++ member), 125  
 safe::event\_t::\_lock (C++ member), 125  
 safe::event\_t::\_status (C++ member), 125  
 safe::event\_t::peek (C++ function), 124  
 safe::event\_t::pop (C++ function), 124  
 safe::event\_t::raise (C++ function), 124  
 safe::event\_t::reset (C++ function), 124  
 safe::event\_t::running (C++ function), 124  
 safe::event\_t::status\_t (C++ type), 124  
 safe::event\_t::stop (C++ function), 124  
 safe::event\_t::view (C++ function), 124  
 safe::lock (C++ function), 123  
 safe::mail\_raw\_t (C++ class), 125  
 safe::mail\_raw\_t::cleanup (C++ function), 125  
 safe::mail\_raw\_t::event (C++ function), 125  
 safe::mail\_raw\_t::event\_t (C++ type), 125  
 safe::mail\_raw\_t::id\_to\_post (C++ member), 125  
 safe::mail\_raw\_t::mutex (C++ member), 125  
 safe::mail\_raw\_t::queue (C++ function), 125  
 safe::mail\_raw\_t::queue\_t (C++ type), 125  
 safe::mail\_t (C++ type), 123  
 safe::make\_alarm (C++ function), 123  
 safe::make\_shared (C++ function), 123  
 safe::post\_t (C++ class), 125  
 safe::post\_t::~~post\_t (C++ function), 126  
 safe::post\_t::mail (C++ member), 126  
 safe::post\_t::post\_t (C++ function), 126  
 safe::queue\_t (C++ class), 126  
 safe::queue\_t::\_continue (C++ member), 126  
 safe::queue\_t::\_cv (C++ member), 126  
 safe::queue\_t::\_lock (C++ member), 126  
 safe::queue\_t::\_max\_elements (C++ member), 126  
 safe::queue\_t::\_queue (C++ member), 126  
 safe::queue\_t::peek (C++ function), 126  
 safe::queue\_t::pop (C++ function), 126  
 safe::queue\_t::queue\_t (C++ function), 126  
 safe::queue\_t::raise (C++ function), 126  
 safe::queue\_t::running (C++ function), 126  
 safe::queue\_t::status\_t (C++ type), 126  
 safe::queue\_t::stop (C++ function), 126  
 safe::queue\_t::unsafe (C++ function), 126  
 safe::shared\_t (C++ class), 127  
 safe::shared\_t::\_construct (C++ member), 127  
 safe::shared\_t::\_count (C++ member), 127  
 safe::shared\_t::\_destruct (C++ member), 127  
 safe::shared\_t::\_lock (C++ member), 127  
 safe::shared\_t::\_object\_buf (C++ member), 127  
 safe::shared\_t::construct\_f (C++ type), 127  
 safe::shared\_t::destruct\_f (C++ type), 127  
 safe::shared\_t::element\_type (C++ type), 127  
 safe::shared\_t::ptr\_t (C++ struct), 127

[safe::shared\\_t::ptr\\_t::~~ptr\\_t \(C++ function\), 128](#)  
[safe::shared\\_t::ptr\\_t::get \(C++ function\), 127](#)  
[safe::shared\\_t::ptr\\_t::operator bool \(C++ function\), 127](#)  
[safe::shared\\_t::ptr\\_t::operator= \(C++ function\), 127](#)  
[safe::shared\\_t::ptr\\_t::operator-> \(C++ function\), 127](#)  
[safe::shared\\_t::ptr\\_t::owner \(C++ member\), 128](#)  
[safe::shared\\_t::ptr\\_t::ptr\\_t \(C++ function\), 128](#)  
[safe::shared\\_t::ptr\\_t::release \(C++ function\), 128](#)  
[safe::shared\\_t::ref \(C++ function\), 127](#)  
[safe::shared\\_t::shared\\_t \(C++ function\), 127](#)  
[safe::signal\\_t \(C++ type\), 123](#)  
[stream \(C++ type\), 117](#)  
[stream::AUDIO\\_STREAM\\_PORT \(C++ member\), 117](#)  
[stream::config\\_t \(C++ struct\), 117](#)  
[stream::config\\_t::audio \(C++ member\), 117](#)  
[stream::config\\_t::audioQosType \(C++ member\), 117](#)  
[stream::config\\_t::controlProtocolType \(C++ member\), 117](#)  
[stream::config\\_t::featureFlags \(C++ member\), 117](#)  
[stream::config\\_t::gcmmap \(C++ member\), 117](#)  
[stream::config\\_t::minRequiredFecPackets \(C++ member\), 117](#)  
[stream::config\\_t::monitor \(C++ member\), 118](#)  
[stream::config\\_t::packetSize \(C++ member\), 118](#)  
[stream::config\\_t::videoQosType \(C++ member\), 118](#)  
[stream::CONTROL\\_PORT \(C++ member\), 117](#)  
[stream::session \(C++ type\), 118](#)  
[stream::session::state\\_e \(C++ enum\), 118](#)  
[stream::session::state\\_e::RUNNING \(C++ enumerator\), 118](#)  
[stream::session::state\\_e::STARTING \(C++ enumerator\), 118](#)  
[stream::session::state\\_e::STOPPED \(C++ enumerator\), 118](#)  
[stream::session::state\\_e::STOPPING \(C++ enumerator\), 118](#)  
[stream::VIDEO\\_STREAM\\_PORT \(C++ member\), 117](#)  
[sync\\_util \(C++ type\), 118](#)  
[sync\\_util::sync\\_t \(C++ class\), 118](#)  
[sync\\_util::sync\\_t::\\_lock \(C++ member\), 119](#)  
[sync\\_util::sync\\_t::lock \(C++ function\), 119](#)  
[sync\\_util::sync\\_t::mutex\\_t \(C++ type\), 118](#)  
[sync\\_util::sync\\_t::operator\\* \(C++ function\), 119](#)  
[sync\\_util::sync\\_t::operator= \(C++ function\), 119](#)  
[sync\\_util::sync\\_t::operator-> \(C++ function\), 119](#)  
[sync\\_util::sync\\_t::raw \(C++ member\), 119](#)  
[sync\\_util::sync\\_t::sync\\_t \(C++ function\), 119](#)  
[sync\\_util::sync\\_t::value\\_t \(C++ type\), 118](#)

## T

[task\\_pool \(C++ member\), 98](#)  
[task\\_pool\\_util \(C++ type\), 119](#)  
[task\\_pool\\_util::\\_Impl \(C++ class\), 119](#)  
[task\\_pool\\_util::\\_Impl::\\_func \(C++ member\), 120](#)  
[task\\_pool\\_util::\\_Impl::\\_Impl \(C++ function\), 120](#)  
[task\\_pool\\_util::\\_Impl::run \(C++ function\), 120](#)  
[task\\_pool\\_util::\\_ImplBase \(C++ class\), 120](#)  
[task\\_pool\\_util::\\_ImplBase::~~ImplBase \(C++ function\), 120](#)  
[task\\_pool\\_util::\\_ImplBase::run \(C++ function\), 120](#)  
[task\\_pool\\_util::TaskPool \(C++ class\), 120](#)  
[task\\_pool\\_util::TaskPool::\\_task \(C++ type\), 120](#)  
[task\\_pool\\_util::TaskPool::\\_time\\_point \(C++ type\), 120](#)  
[task\\_pool\\_util::TaskPool::\\_task\\_mutex \(C++ member\), 121](#)  
[task\\_pool\\_util::TaskPool::\\_tasks \(C++ member\), 121](#)  
[task\\_pool\\_util::TaskPool::\\_timer\\_tasks \(C++ member\), 121](#)  
[task\\_pool\\_util::TaskPool::cancel \(C++ function\), 120](#)  
[task\\_pool\\_util::TaskPool::delay \(C++ function\), 120](#)  
[task\\_pool\\_util::TaskPool::next \(C++ function\), 120](#)  
[task\\_pool\\_util::TaskPool::operator= \(C++ function\), 120](#)  
[task\\_pool\\_util::TaskPool::pop \(C++ function\), 120](#)  
[task\\_pool\\_util::TaskPool::push \(C++ function\), 120](#)  
[task\\_pool\\_util::TaskPool::pushDelayed \(C++ function\), 121](#)  
[task\\_pool\\_util::TaskPool::ready \(C++ function\), 121](#)  
[task\\_pool\\_util::TaskPool::task\\_id\\_t \(C++ type\), 120](#)  
[task\\_pool\\_util::TaskPool::TaskPool \(C++ function\), 121](#)  
[task\\_pool\\_util::TaskPool::timer\\_task\\_t \(C++ class\), 121](#)  
[task\\_pool\\_util::TaskPool::timer\\_task\\_t::future \(C++ member\), 121](#)  
[task\\_pool\\_util::TaskPool::timer\\_task\\_t::task\\_id \(C++ member\), 121](#)

task\_pool\_util::TaskPool::timer\_task\_t::timer\_task\_t  
     (C++ function), 121  
 task\_pool\_util::TaskPool::toRunnable (C++  
     function), 121  
 thread\_pool\_util (C++ type), 122  
 thread\_pool\_util::ThreadPool (C++ class), 122  
 thread\_pool\_util::ThreadPool::\_\_task (C++  
     type), 122  
 thread\_pool\_util::ThreadPool::\_continue  
     (C++ member), 122  
 thread\_pool\_util::ThreadPool::\_cv (C++ mem-  
     ber), 122  
 thread\_pool\_util::ThreadPool::\_lock (C++  
     member), 122  
 thread\_pool\_util::ThreadPool::\_main (C++  
     function), 122  
 thread\_pool\_util::ThreadPool::\_thread (C++  
     member), 122  
 thread\_pool\_util::ThreadPool::~ThreadPool  
     (C++ function), 122  
 thread\_pool\_util::ThreadPool::join (C++ func-  
     tion), 122  
 thread\_pool\_util::ThreadPool::push (C++ func-  
     tion), 122  
 thread\_pool\_util::ThreadPool::pushDelayed  
     (C++ function), 122  
 thread\_pool\_util::ThreadPool::start (C++  
     function), 122  
 thread\_pool\_util::ThreadPool::stop (C++ func-  
     tion), 122  
 thread\_pool\_util::ThreadPool::ThreadPool  
     (C++ function), 122  
 tray\_donate\_github\_cb (C++ function), 97  
 tray\_donate\_mee6\_cb (C++ function), 97  
 tray\_donate\_patreon\_cb (C++ function), 97  
 tray\_donate\_paypal\_cb (C++ function), 97  
 tray\_open\_ui\_cb (C++ function), 97  
 tray\_quit\_cb (C++ function), 97

## U

upnp (C++ type), 128  
 uuid\_util (C++ type), 128  
 uuid\_util::uuid\_t (C++ union), 128  
 uuid\_util::uuid\_t::b16 (C++ member), 129  
 uuid\_util::uuid\_t::b32 (C++ member), 129  
 uuid\_util::uuid\_t::b64 (C++ member), 129  
 uuid\_util::uuid\_t::b8 (C++ member), 129  
 uuid\_util::uuid\_t::generate (C++ function), 129  
 uuid\_util::uuid\_t::operator== (C++ function),  
     128  
 uuid\_util::uuid\_t::operator> (C++ function), 128  
 uuid\_util::uuid\_t::operator< (C++ function), 128  
 uuid\_util::uuid\_t::string (C++ function), 128  
 va (C++ type), 132  
 verbose (C++ member), 98  
 video (C++ type), 129  
 video::color\_t (C++ struct), 129  
 video::color\_t::color\_vec\_u (C++ member), 129  
 video::color\_t::color\_vec\_v (C++ member), 129  
 video::color\_t::color\_vec\_y (C++ member), 129  
 video::color\_t::range\_uv (C++ member), 129  
 video::color\_t::range\_y (C++ member), 130  
 video::config\_t (C++ struct), 130  
 video::config\_t::bitrate (C++ member), 130  
 video::config\_t::dynamicRange (C++ member),  
     130  
 video::config\_t::encoderCscMode (C++ member),  
     130  
 video::config\_t::framerate (C++ member), 130  
 video::config\_t::height (C++ member), 130  
 video::config\_t::numRefFrames (C++ member),  
     130  
 video::config\_t::slicesPerFrame (C++ member),  
     130  
 video::config\_t::videoFormat (C++ member), 130  
 video::config\_t::width (C++ member), 130  
 video::float2 (C++ type), 129  
 video::float3 (C++ type), 129  
 video::float4 (C++ type), 129  
 video::hdr\_info\_raw\_t (C++ struct), 130  
 video::hdr\_info\_raw\_t::enabled (C++ member),  
     130  
 video::hdr\_info\_raw\_t::hdr\_info\_raw\_t (C++  
     function), 130  
 video::hdr\_info\_raw\_t::metadata (C++ member),  
     130  
 video::hdr\_info\_t (C++ type), 129  
 video::packet\_raw\_t (C++ struct), 130  
 video::packet\_raw\_t::~~packet\_raw\_t (C++ func-  
     tion), 131  
 video::packet\_raw\_t::av\_packet (C++ member),  
     131  
 video::packet\_raw\_t::channel\_data (C++ mem-  
     ber), 131  
 video::packet\_raw\_t::init\_packet (C++ func-  
     tion), 131  
 video::packet\_raw\_t::packet\_raw\_t (C++ func-  
     tion), 131  
 video::packet\_raw\_t::replace\_t (C++ struct),  
     131  
 video::packet\_raw\_t::replace\_t::new (C++  
     member), 131  
 video::packet\_raw\_t::replace\_t::old (C++  
     member), 131  
 video::packet\_raw\_t::replace\_t::replace\_t  
     (C++ function), 131

`video::packet_raw_t::replacements` (C++ *member*), [131](#)

`video::packet_t` (C++ *type*), [129](#)

## W

`warning` (C++ *member*), [98](#)

`WEB_DIR` (C *macro*), [106](#)

`wl` (C++ *type*), [132](#)

`wl::monitor_t` (C++ *class*), [132](#)

`wl::monitor_t::description` (C++ *member*), [133](#)

`wl::monitor_t::listen` (C++ *function*), [132](#)

`wl::monitor_t::monitor_t` (C++ *function*), [132](#)

`wl::monitor_t::name` (C++ *member*), [133](#)

`wl::monitor_t::operator=` (C++ *function*), [132](#)

`wl::monitor_t::output` (C++ *member*), [133](#)

`wl::monitor_t::viewport` (C++ *member*), [133](#)

`write_file` (C++ *function*), [97](#)