

[Genymobile](#) / [scrcpy](#)

Join GitHub today

Dismiss

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Display and control your Android device

[#android](#) [#c](#) [#sdl2](#) [#libav](#) [#ffmpeg](#) [#screen](#) [#mirroring](#) [#recording](#)

761 commits

71 branches

0 packages

15 releases

35 contributors

Apache-2.0

Branch: **master** ▾
[New pull request](#)
[Find file](#)
[Clone or download](#) ▾

rom1v Explain master and dev branches in BUILD ...		Latest commit 9b9e717 7 hours ago
.github/ISSUE_TEMPLATE	Add Android device and version in issue template	4 days ago
app	Happy new year 2020!	3 months ago
assets	Update README	2 years ago
config	Fix typos	5 months ago
gradle/wrapper	Uprev Gradle wrapper to latest stable	8 months ago
prebuilt-deps	Upgrade platform-tools (29.0.5) for Windows	5 months ago
scripts	Rename scrcpy-server.jar to scrcpy-server	5 months ago
server	Bump version to 1.12.1	4 months ago
.gitignore	Convert server to an Android project	2 years ago
BUILD.md	Explain master and dev branches in BUILD	7 hours ago
DEVELOP.md	Documentation rectifications	2 months ago
FAQ.ko.md	Add Korean translation for README and FAQ	5 months ago
FAQ.md	Improve "low quality" section in FAQ	7 hours ago
LICENSE	Happy new year 2020!	3 months ago
Makefile.CrossWindows	Upgrade FFmpeg (4.2.1) for Windows	5 months ago
README.ko.md	Happy new year 2020!	3 months ago
README.md	Add scoop instructions for Windows	24 days ago
build.gradle	Uprev AGP to latest stable	8 months ago
cross_win32.txt	Upgrade FFmpeg (4.2.1) for Windows	5 months ago
cross_win64.txt	Upgrade FFmpeg (4.2.1) for Windows	5 months ago
gradle.properties	Enable video output file, with pts set by server	17 months ago
gradlew	Uprev Gradle wrapper to latest stable	8 months ago

gradlew.bat	Uprev Gradle wrapper to latest stable	8 months ago
meson.build	Bump version to 1.12.1	4 months ago
meson_options.txt	Document how to attach a debugger to the server	5 months ago
release.sh	Rename scrcpy-server.jar to scrcpy-server	5 months ago
run	Rename scrcpy-server.jar to scrcpy-server	5 months ago
settings.gradle	Convert server to an Android project	2 years ago

README.md

scrcpy (v1.12.1)

This application provides display and control of Android devices connected on USB (or [over TCP/IP](#)). It does not require any *root* access. It works on *GNU/Linux*, *Windows* and *macOS*.



It focuses on:

- **lightness** (native, displays only the device screen)
- **performance** (30~60fps)
- **quality** (1920×1080 or above)
- **low latency** (35~70ms)
- **low startup time** (~1 second to display the first image)
- **non-intrusiveness** (nothing is left installed on the device)

Requirements

The Android device requires at least API 21 (Android 5.0).

Make sure you [enabled adb debugging](#) on your device(s).

On some devices, you also need to enable [an additional option](#) to control it using keyboard and mouse.

Get the app

Linux

In Debian (*testing* and *sid* for now):

```
apt install scrcpy
```

A [Snap](#) package is available: [scrcpy](#) .

For Arch Linux, an [AUR](#) package is available: [scrcpy](#) .

For Gentoo, an [Ebuild](#) is available: [scrcpy/](#) .

You could also [build the app manually](#) (don't worry, it's not that hard).

Windows

For Windows, for simplicity, a prebuilt archive with all the dependencies (including `adb`) is available:

- [scrcpy-win64-v1.12.1.zip](#)
(SHA-256: `57d34b6d16cfd9fe169bc37c4df58ebd256d05c1ea3febc63d9cb0a027ab47c9`)

It is also available in [Chocolatey](#):

```
choco install scrcpy
choco install adb # if you don't have it yet
```

And in [Scoop](#):

```
scoop install scrcpy
scoop install adb # if you don't have it yet
```

You can also [build the app manually](#).

macOS

The application is available in [Homebrew](#). Just install it:

```
brew install scrcpy
```

You need `adb` , accessible from your `PATH` . If you don't have it yet:

```
brew cask install android-platform-tools
```

You can also [build the app manually](#).

Run

Plug an Android device, and execute:

```
scrcpy
```

It accepts command-line arguments, listed by:

```
scrcpy --help
```

Features

Capture configuration

Reduce size

Sometimes, it is useful to mirror an Android device at a lower definition to increase performance.

To limit both the width and height to some value (e.g. 1024):

```
scrcpy --max-size 1024  
scrcpy -m 1024 # short version
```

The other dimension is computed to that the device aspect ratio is preserved. That way, a device in 1920×1080 will be mirrored at 1024×576.

Change bit-rate

The default bit-rate is 8 Mbps. To change the video bitrate (e.g. to 2 Mbps):

```
scrcpy --bit-rate 2M  
scrcpy -b 2M # short version
```

Limit frame rate

On devices with Android ≥ 10 , the capture frame rate can be limited:

```
scrcpy --max-fps 15
```

Crop

The device screen may be cropped to mirror only part of the screen.

This is useful for example to mirror only one eye of the Oculus Go:

```
scrcpy --crop 1224:1440:0:0 # 1224x1440 at offset (0,0)
```

If `--max-size` is also specified, resizing is applied after cropping.

Recording

It is possible to record the screen while mirroring:

```
scrcpy --record file.mp4  
scrcpy -r file.mkv
```

To disable mirroring while recording:

```
scrcpy --no-display --record file.mp4
scrcpy -Nr file.mkv
# interrupt recording with Ctrl+C
# Ctrl+C does not terminate properly on Windows, so disconnect the device
```

"Skipped frames" are recorded, even if they are not displayed in real time (for performance reasons). Frames are *timestamped* on the device, so [packet delay variation](#) does not impact the recorded file.

Connection

Wireless

Scrcpy uses `adb` to communicate with the device, and `adb` can [connect](#) to a device over TCP/IP:

1. Connect the device to the same Wi-Fi as your computer.
2. Get your device IP address (in Settings → About phone → Status).
3. Enable adb over TCP/IP on your device: `adb tcpip 5555`.
4. Unplug your device.
5. Connect to your device: `adb connect DEVICE_IP:5555` (*replace* `DEVICE_IP`).
6. Run `scrcpy` as usual.

It may be useful to decrease the bit-rate and the definition:

```
scrcpy --bit-rate 2M --max-size 800
scrcpy -b2M -m800 # short version
```

Multi-devices

If several devices are listed in `adb devices`, you must specify the *serial*:

```
scrcpy --serial 0123456789abcdef
scrcpy -s 0123456789abcdef # short version
```

If the device is connected over TCP/IP:

```
scrcpy --serial 192.168.0.1:5555
scrcpy -s 192.168.0.1:5555 # short version
```

You can start several instances of *scrcpy* for several devices.

SSH tunnel

To connect to a remote device, it is possible to connect a local `adb` client to a remote `adb` server (provided they use the same version of the *adb* protocol):

```
adb kill-server # kill the local adb server on 5037
ssh -CN -L5037:localhost:5037 -R27183:localhost:27183 your_remote_computer
# keep this open
```

From another terminal:

```
scrcpy
```

Like for wireless connections, it may be useful to reduce quality:

```
scrcpy -b2M -m800 --max-fps 15
```

Window configuration

Title

By default, the window title is the device model. It can be changed:

```
scrcpy --window-title 'My device'
```

Position and size

The initial window position and size may be specified:

```
scrcpy --window-x 100 --window-y 100 --window-width 800 --window-height 600
```

Borderless

To disable window decorations:

```
scrcpy --window-borderless
```

Always on top

To keep the scrcpy window always on top:

```
scrcpy --always-on-top
```

Fullscreen

The app may be started directly in fullscreen:

```
scrcpy --fullscreen  
scrcpy -f # short version
```

Fullscreen can then be toggled dynamically with `ctrl + f`.

Other mirroring options

Read-only

To disable controls (everything which can interact with the device: input keys, mouse events, drag&drop files):

```
scrcpy --no-control  
scrcpy -n
```

Turn screen off

It is possible to turn the device screen off while mirroring on start with a command-line option:

```
scrcpy --turn-screen-off  
scrcpy -S
```

Or by pressing `Ctrl + o` at any time.

To turn it back on, press `POWER` (or `Ctrl + p`).

Render expired frames

By default, to minimize latency, *scrcpy* always renders the last decoded frame available, and drops any previous one.

To force the rendering of all frames (at a cost of a possible increased latency), use:

```
scrcpy --render-expired-frames
```

Show touches

For presentations, it may be useful to show physical touches (on the physical device).

Android provides this feature in *Developers options*.

Scrcpy provides an option to enable this feature on start and disable on exit:

```
scrcpy --show-touches  
scrcpy -t
```

Note that it only shows *physical* touches (with the finger on the device).

Input control

Rotate device screen

Press `Ctrl + r` to switch between portrait and landscape modes.

Note that it rotates only if the application in foreground supports the requested orientation.

Copy-paste

It is possible to synchronize clipboards between the computer and the device, in both directions:

- `Ctrl + c` copies the device clipboard to the computer clipboard;
- `Ctrl + Shift + v` copies the computer clipboard to the device clipboard;
- `Ctrl + v` pastes the computer clipboard as a sequence of text events (but breaks non-ASCII characters).

Text injection preference

There are two kinds of [events](#) generated when typing text:

- *key events*, signaling that a key is pressed or released;
- *text events*, signaling that a text has been entered.

By default, letters are injected using key events, so that the keyboard behaves as expected in games (typically for WASD keys).

But this may [cause issues](#). If you encounter such a problem, you can avoid it by:

```
scrcpy --prefer-text
```

(but this will break keyboard behavior in games)

File drop

Install APK

To install an APK, drag & drop an APK file (ending with `.apk`) to the `scrcpy` window.

There is no visual feedback, a log is printed to the console.

Push file to device

To push a file to `/sdcard/` on the device, drag & drop a (non-APK) file to the `scrcpy` window.

There is no visual feedback, a log is printed to the console.

The target directory can be changed on start:

```
scrcpy --push-target /sdcard/foo/bar/
```

Audio forwarding

Audio is not forwarded by `scrcpy`. Use [USBaudio](#) (Linux-only).

Also see [issue #14](#).

Shortcuts

Action	Shortcut	Shortcut (macOS)
Switch fullscreen mode	<code>Ctrl + f</code>	<code>Cmd + f</code>
Resize window to 1:1 (pixel-perfect)	<code>Ctrl + g</code>	<code>Cmd + g</code>
Resize window to remove black borders	<code>Ctrl + x</code> <i>Double-click</i> ¹	<code>Cmd + x</code> <i>Double-click</i> ¹
Click on <code>HOME</code>	<code>Ctrl + h</code> <i>Middle-click</i>	<code>Ctrl + h</code> <i>Middle-click</i>
Click on <code>BACK</code>	<code>Ctrl + b</code> <i>Right-click</i> ²	<code>Cmd + b</code> <i>Right-click</i> ²
Click on <code>APP_SWITCH</code>	<code>Ctrl + s</code>	<code>Cmd + s</code>
Click on <code>MENU</code>	<code>Ctrl + m</code>	<code>Ctrl + m</code>
Click on <code>VOLUME_UP</code>	<code>Ctrl + ↑</code> (<i>up</i>)	<code>Cmd + ↑</code> (<i>up</i>)
Click on <code>VOLUME_DOWN</code>	<code>Ctrl + ↓</code> (<i>down</i>)	<code>Cmd + ↓</code> (<i>down</i>)
Click on <code>POWER</code>	<code>Ctrl + p</code>	<code>Cmd + p</code>
Power on	<i>Right-click</i> ²	<i>Right-click</i> ²
Turn device screen off (keep mirroring)	<code>Ctrl + o</code>	<code>Cmd + o</code>
Rotate device screen	<code>Ctrl + r</code>	<code>Cmd + r</code>
Expand notification panel	<code>Ctrl + n</code>	<code>Cmd + n</code>
Collapse notification panel	<code>Ctrl + Shift + n</code>	<code>Cmd + Shift + n</code>
Copy device clipboard to computer	<code>Ctrl + c</code>	<code>Cmd + c</code>
Paste computer clipboard to device	<code>Ctrl + v</code>	<code>Cmd + v</code>
Copy computer clipboard to device	<code>Ctrl + Shift + v</code>	<code>Cmd + Shift + v</code>
Enable/disable FPS counter (on stdout)	<code>Ctrl + i</code>	<code>Cmd + i</code>

¹Double-click on black borders to remove them.

²Right-click turns the screen on if it was off, presses `BACK` otherwise.

Custom paths

To use a specific `adb` binary, configure its path in the environment variable `ADB` :

```
ADB=/path/to/adb scrcpy
```

To override the path of the `scrcpy-server` file, configure its path in `SCRCPY_SERVER_PATH` .

Why *scrcpy*?

A colleague challenged me to find a name as unpronounceable as [gnirehtet](#).

`strcpy` copies a **string**; `scrcpy` copies a **screen**.

How to build?

See [BUILD](#).

Common issues

See the [FAQ](#).

Developers

Read the [developers page](#).

Licence

```
Copyright (C) 2018 Genymobile  
Copyright (C) 2018-2020 Romain Vimont
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.
```

Articles

- [Introducing scrcpy](#)
- [Scrcpy now works wirelessly](#)