Build scrcpy

Here are the instructions to build *scrcpy* (client and server).

Simple

If you just want to install the latest release from `master`, follow this simplified process.

First, you need to install the required packages:

```
# for Debian/Ubuntu
sudo apt install ffmpeg libsdl2-2.0-0 adb wget \
                 gcc git pkg-config meson ninja-build libsdl2-dev \
                 libavcodec-dev libavdevice-dev libavformat-dev libavutil-dev \
                 libusb-1.0-0 libusb-1.0-0-dev
```

Then clone the repo and execute the installation script (source):

```
git clone https://github.com/Genymobile/scrcpy
cd scrcpy
./install_release.sh
```

When a new release is out, update the repo and reinstall:

```
git pull
./install_release.sh
```

To uninstall:

```
sudo ninja -Cbuild-auto uninstall
```

Branches

`master`

The `master` branch concerns the latest release, and is the home page of the project on Github.

`dev`

`dev` is the current development branch. Every commit present in `dev` will be in the next release.

If you want to contribute code, please base your commits on the latest `dev` branch.

Requirements

You need adb. It is available in the Android SDK platform tools, or packaged in your distribution (`adb`).

On Windows, download the platform-tools and extract the following files to a directory accessible from your `PATH` :

- `adb.exe`

- `AdbWinApi.dll`
- `AdbWinUsbApi.dll`

The client requires FFmpeg and LibSDL2. Just follow the instructions.

System-specific steps

Linux

Install the required packages from your package manager.

Debian/Ubuntu

```
# runtime dependencies
sudo apt install ffmpeg libsdl2-2.0-0 adb libusb-1.0-0

# client build dependencies
sudo apt install gcc git pkg-config meson ninja-build libsdl2-dev \
                 libavcodec-dev libavdevice-dev libavformat-dev libavutil-dev \
                 libusb-1.0-0-dev

# server build dependencies
sudo apt install openjdk-11-jdk
```

On old versions (like Ubuntu 16.04), `meson` is too old. In that case, install it from `pip3`:

```
sudo apt install python3-pip
pip3 install meson
```

Fedora

```
# enable RPM fusion free
sudo dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-
release-$(rpm -E %fedora).noarch.rpm

# client build dependencies
sudo dnf install SDL2-devel ffms2-devel libusb-devel meson gcc make

# server build dependencies
sudo dnf install java-devel
```

Windows

Cross-compile from Linux

This is the preferred method (and the way the release is built).

From *Debian*, install *mingw*:

```
sudo apt install mingw-w64 mingw-w64-tools
```

You also need the JDK to build the server:

```
sudo apt install openjdk-11-jdk
```

Then generate the releases:

```
./release.sh
```

It will generate win32 and win64 releases into `dist/`.

In MSYS2

From Windows, you need MSYS2 to build the project. From an MSYS2 terminal, install the required packages:

```
# runtime dependencies
pacman -S mingw-w64-x86_64-SDL2 \
          mingw-w64-x86_64-ffmpeg \
          mingw-w64-x86_64-libusb

# client build dependencies
pacman -S mingw-w64-x86_64-make \
          mingw-w64-x86_64-gcc \
          mingw-w64-x86_64-pkg-config \
          mingw-w64-x86_64-meson
```

For a 32 bits version, replace `x86_64` by `i686`:

```
# runtime dependencies
pacman -S mingw-w64-i686-SDL2 \
          mingw-w64-i686-ffmpeg \
          mingw-w64-i686-libusb

# client build dependencies
pacman -S mingw-w64-i686-make \
          mingw-w64-i686-gcc \
          mingw-w64-i686-pkg-config \
          mingw-w64-i686-meson
```

Java (>= 7) is not available in MSYS2, so if you plan to build the server, install it manually and make it available from the `PATH`:

```
export PATH="$JAVA_HOME/bin:$PATH"
```

Mac OS

Install the packages with Homebrew:

```
# runtime dependencies
brew install sdl2 ffmpeg libusb
```

```
# client build dependencies
brew install pkg-config meson
```

Additionally, if you want to build the server, install Java 8 from Caskroom, and make it available from the `PATH`:

```
brew tap homebrew/cask-versions
brew install adoptopenjdk/openjdk/adoptopenjdk11
export JAVA_HOME="$(/usr/libexec/java_home --version 1.11)"
export PATH="$JAVA_HOME/bin:$PATH"
```

Docker

See pierlon/scrcpy-docker.

Common steps

As a non-root user, clone the project:

```
git clone https://github.com/Genymobile/scrcpy
cd scrcpy
```

Build

You may want to build only the client: the server binary, which will be pushed to the Android device, does not depend on your system and architecture. In that case, use the prebuilt server (so you will not need Java or the Android SDK).

Option 1: Build everything from sources

Install the Android SDK (*Android Studio*), and set `ANDROID_SDK_ROOT` to its directory. For example:

```
# Linux
export ANDROID_SDK_ROOT=~/Android/Sdk
# Mac
export ANDROID_SDK_ROOT=~/Library/Android/sdk
# Windows
set ANDROID_SDK_ROOT=%LOCALAPPDATA%\Android\sdk
```

Then, build:

```
meson x --buildtype=release --strip -Db_lto=true
ninja -Cx  # DO NOT RUN AS ROOT
```

Note: `ninja` must be run as a non-root user (only `ninja install` must be run as root).

Option 2: Use prebuilt server

- `scrcpy-server-v1.23`
  (SHA-256: 2a913fd47478c0b306fca507cb0beb625e49a19ff9fc7ab904e36ef5b9fe7e68)

Download the prebuilt server somewhere, and specify its path during the Meson configuration:

```
meson x --buildtype=release --strip -Db_lto=true \
    -Dprebuilt_server=/path/to/scrcpy-server
ninja -Cx  # DO NOT RUN AS ROOT
```

The server only works with a matching client version (this server works with the `master` branch).

Run without installing:

```
./run x [options]
```

Install

After a successful build, you can install *scrcpy* on the system:

```
sudo ninja -Cx install    # without sudo on Windows
```

This installs three files:

- `/usr/local/bin/scrcpy`
- `/usr/local/share/scrcpy/scrcpy-server`
- `/usr/local/share/man/man1/scrcpy.1`

You can then run *scrcpy*.

Uninstall

```
sudo ninja -Cx uninstall  # without sudo on Windows
```