
Chromebooks Developer Cookbook

Documentation

Release 1.0

Enric Balletbo i Serra

Jun 27, 2023

Contents:

1	About the Chromebooks Developer Cookbook	1
2	Chromebook developer tool	3
2.1	Switch to developer mode	3
2.2	Enable booting from external storage	3
2.3	Create a USB or SD for dual booting	4
3	GNOME3 Rootfs Demo	5
3.1	The Debian way to build the demo rootfs	5
3.2	Appendix	6
4	Weston Rockchip Demo	9
4.1	Supported and tested hardware	9
4.2	The Debian way to build the demo rootfs	10
4.3	Quick steps to create a SD-card	11
4.4	Connect the Wiimote	11
4.5	Known issues	11
4.6	Appendix	11
5	Indices and tables	13

CHAPTER 1

About the Chromebooks Developer Cookbook

This cookbook is a collection of useful scripts and information to create Debian-based images with the latest kernel that can be flashed on different Chromebook models.

The documentation, like the chromebook developer tool itself, is very much a work in progress. Please note that improvements to the documentation are welcome; create a github account and fork the project if you want to help out.

The latest version of the ‘Chromebooks Developer Cookbook’ can be found here:

- <https://chromebooks.readthedocs.io/en/latest/>

CHAPTER 2

Chromebook developer tool

These instructions will create a dual-booting environment where you can switch between booting Debian and the stock ChromeOS. No changes are made to the internal eMMC drive, and your new Debian install will run completely from external storage. This is the recommended setup for those that just want to take a test drive, or don't want to give up ChromeOS.

You must be running the latest ChromeOS prior to installation.

2.1 Switch to developer mode

1. Turn off the laptop.
2. To invoke Recovery mode, you hold down the ESC and Refresh keys and poke the Power button.
3. At the Recovery screen press Ctrl-D (there's no prompt - you have to know to do it).
4. Confirm switching to developer mode by pressing enter, and the laptop will reboot and reset the system. This takes about 10-15 minutes.

Note: After enabling developer mode, you will need to press Ctrl-D each time you boot, or wait 30 seconds to continue booting.

2.2 Enable booting from external storage

1. After booting into developer mode, hold Ctrl and Alt and poke the F2 key. This will open up the developer console.
2. Type root to the login screen.
3. Then type this to enable USB booting:

```
enable_dev_usb_boot
```

1. Reboot the system to allow the change to take effect.

2.3 Create a USB or SD for dual booting

```
./chromebook-setup.sh help
```

Examples:

1. Create a bootable media for supported aarch64 Chromebooks using a Fedora Workstation Rawhide image and install a minimal custom build of the kernel in the image.

```
./chromebook-setup.sh do_everything --architecture=arm64 --storage=/dev/[sdX|mmcblkX]
```

1. Create a bootable media for supported aarch64 Chromebooks using Fedora Workstation 38

```
tools/image-installer/fedora-workstation-38-aarch64 /dev/[sdX|mmcblkX]
```

Under the hood, this script downloads an uncompress a Fedora-Workstation-38-1.6 image and runs:

```
./chromebook-setup.sh deploy_fedora --architecture=arm64 --storage=/dev/[sdX|mmcblkX] \
--distro=fedora \
--image=Fedora-Workstation-38-1.6.aarch64.raw \
--kparams="clk_ignore_unused deferred_probe_timeout=30"
```

CHAPTER 3

GNOME3 Rootfs Demo

This is the rootfs used by default to build the images for the Chromebooks. The main purpose of this rootfs is show current support in mainline for different boards. The demo runs a *Debian* based image with GNOME3 and accelerated graphics when possible.

The *Debian* images are assembled using the `debos` utility, which uses the *Debian* package feed beneath. Stuff not available in official *Debian* packages will be built from sources or downloaded into the final image.

3.1 The Debian way to build the demo rootfs

To install `debos` you can do the following steps:

```
$ sudo apt install golang git libglib2.0-dev libostree-dev qemu-system-x86 qemu-user-static debootstrap systemd-container xz-utils bmap-tools
$ export GOPATH=`pwd`/gocode
$ go get -u github.com/go-debos/debos/cmd/debos
```

First, make sure you have KVM installed:

```
$ sudo apt install qemu-kvm ovmf
And then run:
$ $GOPATH/bin/debos -m 4G <debos-image.yaml>
```

Now that debos is installed, let's create the demos images, run:

```
Export the architecture of your device (arm or arm64 or amd64):
$ export architecture=arm64
And then run:
```

(continues on next page)

(continued from previous page)

```
$ $GOPATH/bin/debos -m 4G -t architecture:$architecture tools/debos/images/gnome-  
→desktop/gnome-desktop.yaml
```

Will create the following output:

- debian-gnome-desktop-sid-\$architecture.tar.gz, a tarball with the Debian GNOME3 based filesystem.

After that, to create an image run (**note** this is only supported for arm and arm64 architectures):

```
$ $GOPATH/bin/debos -m 4G -t architecture:$architecture tools/debos/images/gnome-  
→desktop/chromebook-image.yaml
```

Will create the following output:

- debian-gnome-desktop-sid-\$architecture.img.gz, a gz-compressed image file for a Chromebook.
- debian-gnome-desktop-sid-\$architecture.img.gz.md5, the image checksum.
- debian-gnome-desktop-sid-\$architecture.img.bmap, a bitmap summary for faster flashing via bmaptools.

To flash it, assuming your SD card is /dev/mmcblk0, use:

```
$ bmaptool copy debian-gnome-desktop-sid-$architecture.img.gz /dev/mmcblk0
```

The bmap file is automatically looked for in the current directory.

Note that the credentials to login are debian:debian.

3.2 Appendix

3.2.1 The Docker way to build the demo rootfs

NOT TESTED YET

This is really simple as an official container is provided for it:

```
$ docker pull godebos/debos
```

To build the image run:

```
$ docker run --rm --interactive --tty --device /dev/kvm --user $(id -u) --workdir /  
→recipes --mount "type=bind,source=$pwd,destination=/recipes" --security-opt  
→label=disable godebos/debos <debos-image.yaml>
```

3.2.2 Extend the rootfs partition to fill available space

What you need to do is to run growpart against you physical disk and partition.

For example your disk is /dev/mmcblk0 and your partition is 2. What growpart does is actually extending the size of your partition to the maximum allowed physical disk size.

You can run below command:

```
$ growpart /dev/mmcblk0 2
```

Once it finish you will see growpart has extend your partition table to the maximum available disk size.

Now, you need to reboot your machine. When the machine comes live again, you can run resize2fs command to extend the filesystem. Below is the sample command.

```
$ resize2fs /dev/mmcblk0p2
```


CHAPTER 4

Weston Rockchip Demo

The purpose of this demo is show current support in mainline for different Rockchip boards. The demo runs a *Debian* based image with Weston and accelerated graphics using Panfrost.

The *Debian* images are assembled using the `debos` utility, which uses the *Debian* package feed beneath. Stuff not available in official *Debian* packages will be built from sources or downloaded into the final image.

4.1 Supported and tested hardware

4.1.1 Samsung Chromebook Plus (kevin)

4.1.2 Samsung Chromebook Plus (kevin)

The Samsung Chromebook Plus is a convertible touchscreen laptop powered by an ARMv8 Rockchip RK3399 hexa-core processor and 4GB RAM, measuring 11.04" x 8.72" x 0.55" and weighing 2.38 lbs.

Features:

- Rockchip RK3399 (OP1) dual-core 2.0GHz Cortex-A72 and quad-core 1.4GHz Cortex-A53 processor
- 4GB LPDDR3 RAM
- 12.3" 2400x1600 LED display
- Mali T860MP4 GPU
- 32GB eMMC
- 5140 mAh battery
- 2x USB 3.0 Type-C ports
- Built-in stylus

4.1.3 ASUS Chromebook Flip C100P (veyron_minnie)

4.1.4 ASUS Chromebook Flip C100P (veyron_minnie)

The ASUS Chromebook Flip C100P is a convertible touchscreen laptop powered by an ARMv7 Rockchip RK3288 processor and 4GB RAM, measuring 262.8 x 182.4 x 15.6 mm (WxDxH) and weighing 0.89 kg.

Features:

- Rockchip RK3288 1.8GHz
- 2GB/4GB LPDDR3 RAM
- 10,1" - 25,65 cm 16:10 WXGA (1280x800) LED display
- Mali T764 GPU
- 16GB/32GB eMMC
- 2Cells 31 Whrs nattery
- 2x USB 2.0 ports

4.2 The Debian way to build the demo rootfs

To install `debos` you can do the following steps:

```
$ sudo apt install golang git libglib2.0-dev libostree-dev qemu-system-x86 qemu-user-static debootstrap systemd-container xz-utils bmap-tools
$ export GOPATH=`pwd`/gocode
$ go get -u github.com/go-debos/debos/cmd/debos
```

First, make sure you have KVM installed:

```
$ sudo apt install qemu-kvm ovmf

And then run:

$ $GOPATH/bin/debos -m 4G <debos-image.yaml>
```

Now that debos is installed, let's create the demos images, run:

```
Export the architecture of your device:

$ export architecture=arm64
or
$ export architecture=armhf

And then run:
$ $GOPATH/bin/debos -m 4G -t architecture:$architecture tools/debos/images/weston-desktop/weston-desktop.yaml
```

Will create the following output:

- `debian-weston-desktop-sid-$architecture.tar.gz`, a tarball with the debian weston based filesystem.

4.3 Quick steps to create a SD-card

```
$ ./chromebook-setup.sh do_everything --architecture=<arm64|arm> --storage=/dev/
↳mmcblkX
```

4.4 Connect the Wiimote

First you need to make sure to load the uinput module:

```
$ modprobe uinput
```

Thanks to cwiid you can scan for your Wiimote now:

```
(press the 1 and 2 buttons on your Wiimote)
$ bluetoothctl scan on
Scanning ...
<MAC address>      Nintendo RVL-CNT-01
```

The Wiimote can act as a regular input device like a mouse using wminput, simply run:

```
$ wminput -w
```

4.5 Known issues

4.5.1 1. Ethernet is down by default

For some reason the ethernet interface (through a docking USB-C) is down, to enable it run:

```
$ ip link set enx0050b6213e94 up
```

4.6 Appendix

4.6.1 The Docker way to build the demo rootfs

NOT TESTED YET

This is really simple as an official container is provided for it:

```
$ docker pull godebos/debos
```

To build the image run:

```
$ docker run --rm --interactive --tty --device /dev/kvm --user $(id -u) --workdir /
↳recipes --mount "type=bind,source=$PWD,destination=/recipes" --security-opt=
↳label=disable godebos/debos <debos-image.yaml>
```


CHAPTER 5

Indices and tables

- genindex
- modindex
- search