

---

# **live-wrapper Documentation**

***Release 0.4***

**Iain R. Learmonth**

October 29, 2016



<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Quickstart</b>	<b>5</b>
2.1	Super Fast Quickstart . . . . .	5
2.2	Customising the Image . . . . .	5
2.3	Testing the Image with QEMU . . . . .	6
2.4	Next Steps . . . . .	6
<b>3</b>	<b>Advanced Topics</b>	<b>7</b>
3.1	Bootloader Customisation . . . . .	7
<b>4</b>	<b>Developer Documentation</b>	<b>9</b>
4.1	Helper Documentation . . . . .	9
<b>5</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>



Contents:



---

# Introduction

---

live-wrapper is a tool produced by the [Debian Live Team](#) that can be used to create Debian-based live images for use with CDs, DVDs or USB sticks.

live-wrapper is designed to, as far as is possible, exist only as glue between existing tools. The root filesystem is built using *vmdebootstrap* and isohybrid image creation is provided by *xorriso*.





---

## Quickstart

---

### 2.1 Super Fast Quickstart

Building images with live-wrapper is quite simple. For the impatient:

```
sudo apt install live-wrapper
sudo lwr
```

This will build you a file named `output.iso` in the current directory containing a minimal live-image.

**Warning:** Currently live-wrapper will create a lot of files and directories in the current working directory. There is a TODO item to move these to a temporary location and clean up afterwards, though this has not yet been fully implemented. You may want to use an empty directory to run `lwr` in.

### 2.2 Customising the Image

There are a number of supported command-line arguments that can be passed to live-wrapper. These change the behaviour to create a customised image.

#### 2.2.1 Changing the Distribution

By default, the ISO image will be built using the `stable` distribution. If you'd like to build using `testing` or `unstable` you can pass the `-d` parameter to live-wrapper like so:

```
sudo lwr -d testing
```

#### 2.2.2 Using an Alternative Mirror

By default, `vmdebootstrap` will use the mirror configured in your `/etc/apt/sources.list`. If you have a faster mirror available, you may want to change the mirror you're using to create the image. You can do this with the `-m` parameter:

```
sudo lwr -m http://localhost/debian/
```

### 2.2.3 Customising Packages

There are two methods of specifying extra packages to be installed into the live image: the `-t` and the `-e` parameters. The difference between these two parameters is that the list of tasks given to `-e` is passed to `debootstrap` for installation as part of the initial root filesystem creation, whereas the packages passed to `-t` are installed as part of the `vmdebootstrap` hook.

This essentially means that any packages installed using `-e` will *not* have their “Recommends” installed, but will have their “Depends” installed while packages installed using `-t` will have both installed making `-t` the suitable place for the installation of task packages.

There is no reason you cannot pass your entire package list to `-t`, these are separated mainly to help with the readability of parameters passed to live-wrapper.

For example:

```
sudo lwr -e vim -t science-typesetting
```

## 2.3 Testing the Image with QEMU

You can easily test your created live images with QEMU.

**Warning:** You will need to increase the amount of memory available to QEMU when running the live image. The image will crash if run with the default memory limit.

To test the image using BIOS boot:

```
qemu-system-x86_64 -m 2G -cdrom live.iso
```

For EFI boot you will need to install the `ovmf` package and then run:

```
qemu-system-x86_64 -bios /usr/share/ovmf/OVMF.fd -m 2G -cdrom live.iso
```

## 2.4 Next Steps

To learn more about using live-wrapper, you can read the man page or check out the [Advanced Topics](#) section of this documentation.

---

## Advanced Topics

---

### 3.1 Bootloader Customisation

By default, both ISOLINUX (for BIOS boot) and GRUB 2 (for EFI boot) are installed into the images. A `loopback.cfg` file is also installed to allow for the image to be booted while still an ISO image in a filesystem.

You can pass `--no-isolinux` to prevent the installation of ISOLINUX to the image if you do not want ISOLINUX.

You can pass `--no-grub` to prevent the installation of GRUB to the image.

If you do not want GRUB installed but you would still like the `loopback.cfg` file to be installed, you can pass `--grub-loopback-only`.



---

## Developer Documentation

---

### 4.1 Helper Documentation

#### 4.1.1 Root Filesystem Helpers

#### 4.1.2 Bootloader Helpers

#### 4.1.3 CD Filesystem Helpers

This module provides helpers for generating the metadata stored in `.disk/` on the `cdroot`. Directory listing of `/.disk/`

```
-r--r--r-- 1 0 0 29 Sep 8 2015 [ 194732 00] archive_trace -r--r--r-- 1 0 0 5 Sep 8 2015 [ 194733 00] base_components  
-r--r--r-- 1 0 0 0 Sep 8 2015 [ 907 00] base_installable -r--r--r-- 1 0 0 5 Sep 8 2015 [ 194734 00] cd_type -r--r--r-- 1 0 0 89  
Sep 8 2015 [ 194735 00] info -r--r--r-- 1 0 0 49 Sep 8 2015 [ 194736 00] udeb_include
```

```
lwr.disk.install_disk_info(cdroot)
```

This function creates the `.disk/info` metadata and installs it into the specified `cdroot`.

#### 4.1.4 ISO Creation Helpers



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`





|

`lwr.disk`, 9



## I

`install_disk_info()` (in module `lwr.disk`), [9](#)

## L

`lwr.disk` (module), [9](#)