<https://willhaley.com/blog/custom-debian-live-environment/>

tftp/dnsmasq

https://www.tecmint.com/install-pxe-network-boot-server-in-centos-7/

[Will Haley](https://willhaley.com/)

[About](https://willhaley.com/about/) [Clippings](https://willhaley.com/clippings/) [Favorites](https://willhaley.com/favorites/) [CV](https://willhaley.com/cv/)

Create a Custom Debian Live Environment (CD or USB)

Feb 8, 2018*Updated Feb 26, 2018*

These are steps that I used on an **Ubuntu 16.04.3 LTS (Xenial Xerus)** 64-bit system to build an **x86 Debian 9 (Stretch)** live environment that I can boot from CD or USB

*This article is periodically updated. So older user comments below may no longer be valid.*

See other related articles here:

* [Perform a Custom Debian Hard Drive Install](https://willhaley.com/blog/custom-debian-hard-drive-install/)
* [Create a Custom Hybrid (Mac or PC) Debian Live USB](https://willhaley.com/blog/create-custom-hybrid-debian-live-environment/)
* [Install Debian to USB Drive](https://willhaley.com/blog/install-debian-usb/)

**Warning**: I have **highlighted** all the places you should be in the chroot environment. Be careful! Running some of these commands on your local environment instead of in the chroot can damage your system.

1. Install applications we need to build the environment.

sudo apt-get install \

debootstrap \

syslinux \

isolinux \

squashfs-tools \

genisoimage \

memtest86+

1. Create a directory where we will store all of our files for creating the live environment.

mkdir $HOME/live\_boot

1. Set up the base Debian environment. I am using stretch for my distribution and i386 for the architecture. *Please change your mirror if you are not in the United States or if you know of a mirror closer to you.* Consult the list of [debian mirrors](https://www.debian.org/mirror/list).

sudo debootstrap \

--arch=i386 \

--variant=minbase \

stretch $HOME/live\_boot/chroot \

http://ftp.us.debian.org/debian/

1. Chroot to our Debian environment.

sudo chroot $HOME/live\_boot/chroot

1. **chroot**

Set a custom hostname for your Debian environment.

echo "debian-live" > /etc/hostname

1. **chroot**

Figure out which Linux Kernel you want in your live environment.

apt-cache search linux-image

1. **chroot**

I chose the image linux-image-686. I also believe live-boot is a necessity. systemd-sys (or an equivalent) is also necessary to provide init.

apt-get update && \

apt-get install --no-install-recommends \

linux-image-686 \

live-boot \

systemd-sysv

1. **chroot**

Install programs of your choosing, and then run apt-get clean to save some space. I use --no-install-recommends to avoid superfluous packages. You should decide what you need for your environment.

Read Debian’s [ReduceDebian article](https://wiki.debian.org/ReduceDebian) for tips on minimizing the size of your Debian environment if size is important.

apt-get install --no-install-recommends \

network-manager net-tools\

tcpdump wget openssh-client \

pciutils rsync lldpad vim iputils-arping iputils-ping \

syslinux lldpd dmidecode && \

apt-get clean

apt-get install --no-install-recommends \

network-manager net-tools wireless-tools wpagui \

tcpdump wget openssh-client \

blackbox xserver-xorg-core xserver-xorg xinit xterm \

pciutils usbutils gparted ntfs-3g hfsprogs rsync dosfstools \

syslinux partclone nano pv \

rtorrent iceweasel chntpw && \

apt-get clean

1. **chroot**

Set the root password (root will be the only user in this live environment)

passwd root

My additions

vi [/lib/systemd/system/getty@.service](mailto:/lib/systemd/system/getty@.service)

Change

ExecStart=-/sbin/agetty --noclear %I $TERM

To ExecStart=-/sbin/agetty --noclear -a root %I $TERM

Add to /etc/ssh/sshd\_config

PermitRootLogin yes

PasswordAuthentication yes

Uncomment HostKey /etc/ssh/ssh\_host\_rsa\_key

1. **chroot**

exit

1. Make directories that will be copied to our bootable medium.

mkdir -p $HOME/live\_boot/image/{live,isolinux}

1. Compress the chroot environment into a Squash filesystem.

(cd $HOME/live\_boot && \

sudo mksquashfs chroot image/live/filesystem.squashfs -e boot

)

1. Prepare our USB/CD bootloader. You should be able to copy and paste these lines into a terminal to save you some time.

(cd $HOME/live\_boot && \

cp chroot/boot/vmlinuz-4.9.0-6-686 image/live/vmlinuz1

cp chroot/boot/initrd.img-4.9.0-6-686 image/live/initrd1

)

1. Create a menu for the isolinux bootloader. Create a text file at $HOME/live\_boot/image/isolinux/isolinux.cfg with this content.

UI menu.c32

prompt 0

menu title Debian Live

timeout 300

label Debian Live 4.9.0-3-686

menu label ^Debian Live 4.9.0-3-686

menu default

kernel /live/vmlinuz1

append initrd=/live/initrd1 boot=live

label hdt

menu label ^Hardware Detection Tool (HDT)

kernel hdt.c32

text help

HDT displays low-level information about the systems hardware.

endtext

label memtest86+

menu label ^Memory Failure Detection (memtest86+)

kernel /live/memtest

scp -r \* root@10.234.122.102:/var/lib/tftpboot/deblive/

Create a bootable medium

CD

Copy files necessary for the ISO to boot and then create the ISO

(cd $HOME/live\_boot/image/ && \

cp /usr/lib/ISOLINUX/isolinux.bin isolinux/ && \

cp /usr/lib/syslinux/modules/bios/menu.c32 isolinux/ && \

cp /usr/lib/syslinux/modules/bios/hdt.c32 isolinux/ && \

cp /usr/lib/syslinux/modules/bios/ldlinux.c32 isolinux/ && \

cp /usr/lib/syslinux/modules/bios/libutil.c32 isolinux/ && \

cp /usr/lib/syslinux/modules/bios/libmenu.c32 isolinux/ && \

cp /usr/lib/syslinux/modules/bios/libcom32.c32 isolinux/ && \

cp /usr/lib/syslinux/modules/bios/libgpl.c32 isolinux/ && \

cp /usr/share/misc/pci.ids isolinux/ && \

cp /boot/memtest86+.bin live/memtest

)

genisoimage \

-rational-rock \

-volid "Debian Live" \

-cache-inodes \

-joliet \

-hfs \

-full-iso9660-filenames \

-b isolinux/isolinux.bin \

-c isolinux/boot.cat \

-no-emul-boot \

-boot-load-size 4 \

-boot-info-table \

-output $HOME/live\_boot/debian-live.iso \

$HOME/live\_boot/image

Now burn the ISO to a CD and you should be ready to boot from it and go.

USB

Copy files necessary for the USB to boot and copy the environment to the USB drive (I am assuming you have an umounted **FAT32** formatted USB drive **/dev/sdz** and the **BOOT** flag is set on /dev/sdz1 and you have a ready mount point at **/mnt/usb**)

sudo syslinux -i /dev/sdz1

sudo dd \

if=/usr/lib/syslinux/mbr.bin \

of=/dev/sdz \

conv=notrunc bs=440 count=1

sudo mount /dev/sdz1 /mnt/usb

You should be able to copy and paste this block into a terminal to save you some time.

sudo cp /usr/lib/syslinux/modules/bios/menu.c32 /mnt/usb/ && \

sudo cp /usr/lib/syslinux/modules/bios/hdt.c32 /mnt/usb/ && \

sudo cp /usr/lib/syslinux/modules/bios/ldlinux.c32 /mnt/usb/ && \

sudo cp /usr/lib/syslinux/modules/bios/libutil.c32 /mnt/usb/ && \

sudo cp /usr/lib/syslinux/modules/bios/libmenu.c32 /mnt/usb/ && \

sudo cp /usr/lib/syslinux/modules/bios/libcom32.c32 /mnt/usb/ && \

sudo cp /usr/lib/syslinux/modules/bios/libgpl.c32 /mnt/usb/ && \

sudo cp /boot/memtest86+.bin /mnt/usb/memtest && \

sudo cp $HOME/live\_boot/image/isolinux/isolinux.cfg /mnt/usb/syslinux.cfg && \

sudo cp /usr/share/misc/pci.ids /mnt/usb/ && \

sudo cp -r $HOME/live\_boot/image/live /mnt/usb/

Now unmount the drive and you should be ready to boot from it and go.

Will Haley

* Will Haley
* [williamhaley](https://github.com/williamhaley)