help.ubuntu.com

LiveCDCustomizationFromScratch - Community Help Wiki

25-32 minutes

This procedure works and can create a bootable Ubuntu LiveCd (along with the automatic hardware detection and configuration) from scratch. You do not need to start from a pre-existing LiveCd.

You may wish to create an Ubuntu Remix and distribute it as a LiveCd. Here is a way to do that without having to start from an existing Ubuntu Desktop Cd.

There are three different areas to think about; the host system, the disk image and the chroot.

The Host System

This refers to the Ubuntu desktop you are running, the one the customised LiveCd is being built on. You will need to install the syslinux, squashfs-tools and genisoimage packages to be able to build the LiveCd Remix using the current system.

The Disk Image

The disk image is a folder that we will burn to Cd. Just create a new folder to be the Disk-Image-Folder. The isolinux bootloader binary (taken from the syslinux package) needs to be copied onto the disk image so it will go into the disk image folder for now. The isolinux configuration file, which will allow the Cd to show a boot-menu at boot time, needs to be copied into there too. You will also copy the kernel from the chroot onto the disk image (folder).

The disk image will be created in the host environment, outside of the chroot.

The ChRoot Environment

This is the system that will eventually run from the disk. It does not need a kernel, nor a boot-loader unless you are planning on installing it back onto a hard disk (using Ubiquity). The Casper package needs to be installed into the chroot. Casper is what allows the Live System to perform hardware autoconfiguration and run from a live environment. Installing the Casper package will update the kernel's initrd to perform these tasks. The kernel that is installed into the chroot will be copied out from the chroot and put into the disk image.

The chroot will end up inside the disk image in the form of a squashed (compressed) file. For right now, it will be just another folder on your host system.

The basic steps are to

1. Create a chroot and install your packages there.
2. Compress the chroot system into a file.
3. Create and configure the disk image which will have the bootloader (isolinux), the kernel, the compressed file-system image and some other stuff.
4. Burn the Cd and test it out.

From a command-line install debootstrap on the host system. Then make a new folder "work" and inside that make another directory "chroot". Then run debootstrap

Note: The version of Debootstrap for a release of ubuntu does not contain the files to bootstrap the next Ubuntu release. For example, you cannot bootstrap Karmic on an Intrepid system without intalling Karmic's debootstrap package. Find the version of debootstrap you need here and install it using dpkg. The debootstrap package doesn't depend on any other packages and so installing it "by hand" will not cause any problems on your system.

```
sudo apt-get install debootstrap

# alternatively, download the package and run
sudo dpkg -i debootstrap_$VERSION.deb

mkdir -p work/chroot
```

```
cd work

sudo debootstrap --arch=$ARCH $RELEASE chroot
```

$VERSION above should be replaced with whatever you have obtained. Likewise, $RELEASE is the version of Ubuntu you intend to build an ISO for. $ARCH is the target processor architecture. For old 32 bit x86 systems use i386. For newer 64-bit x86 systems (also known as x64, x86_64, Intel 64, and AMD64) use amd64 (--arch=amd64).

The code above creates a directory called work, with a chroot directory inside it. The debootstrap command installs a bare Ubuntu system into work/chroot.

If downloading from the main archive is slow, use one of the alternatives from this list of mirrors by adding the URL to the end of the debootstrap command, otherwise the ubuntu.com archive will be used by default.

Note: If you want to build a newer release of Ubuntu which you cannot bootstrap, for example oneiric:

```
cd /usr/share/debootstrap/scripts/
sudo ln -s gutsy oneiric
cd
mkdir -p work/chroot

cd work
sudo debootstrap --arch=amd64 oneiric chroot
```

It is important to install custom applications such as MySQL after linux-generic is installed because such applications require kernel modules for post-install configurations.

If you are planning on installing anything using the package desktop-base (xfce4 for instance), you will also need to bind your /dev to the chroot as well (not just devpts). Otherwise, grube-probe will error out and you won't be able to finish the installations. Replace /path/to/chroot/dev with your respective chroot.

```
sudo mount --bind /dev chroot/dev
```

Now copy the system files so you can get some internet in the chroot.

```
sudo cp /etc/hosts chroot/etc/hosts
sudo cp /etc/resolv.conf chroot/etc/resolv.conf
sudo cp /etc/apt/sources.list chroot/etc
/apt/sources.list
```

Note: If you are bootstrapping a release of Ubuntu other then the release you are currently running you should substitute the 'sudo cp /etc/apt/sources.list chroot/etc/apt/sources.list' command with the following.

```
sudo sed s/<Release-You-Are-On>/<Release-You-Are-
Bootstrapping>/ < /etc/apt/sources.list >
chroot/etc/apt/sources.list
```

For example if you are running precise and you are bootstrapping oneiric the command would be:

```
sudo sed s/precise/oneiric/ < /etc/apt
/sources.list > chroot/etc/apt/sources.list
```

You may edit the sources.list in the chroot to add a line from a PPA, if you need. You will need to add the PPA's key to your chroot's package manager. On the PPA's overview page you'll see the PPA's OpenPGP key id. It'll look something like this: 1024/12345678. Copy it, or make a note of, the portion after the slash, e.g: 12345678. This key will be added once we enter the chroot.

Important: Make a backup copy of /sbin/initctl this next step will delete this file. There is a problem with 10.04 upstart package not containing /sbin/initctl.distrib and even after you update upstart the directions for leaving the chroot do not seem to restore this file.

```
sudo chroot chroot

mount none -t proc /proc
mount none -t sysfs /sys
mount none -t devpts /dev/pts
export HOME=/root
export LC_ALL=C
```

```
sudo apt-key adv --keyserver keyserver.ubuntu.com
--recv-keys 12345678  #Substitute "12345678" with
the PPA's OpenPGP ID.
apt-get update
apt-get install --yes dbus
dbus-uuidgen > /var/lib/dbus/machine-id
dpkg-divert --local --rename --add /sbin/initctl
```

There is a current (for Karmic, Lucid, ..., Precise) issue with services running in a chroot: https://bugs.launchpad.net/ubuntu/+source/upstart/+bug/430224.

A workaround is to link /sbin/initctl to /bin/true.

```
ln -s /bin/true /sbin/initctl
```

Upgrade packages if you want:

```
apt-get --yes upgrade
```

Install packages needed for Live System:

```
apt-get install --yes ubuntu-standard casper
lupin-casper
apt-get install --yes discover laptop-detect os-
prober
apt-get install --yes linux-generic
```

Before Maverick, discover named to discover1. Adjust the preceding lines accordingly.

If you make Lucid Lynx (10.04) base Live system you need install grub2 plymouth-x11 packages:

```
apt-get install --yes grub2 plymouth-x11
```

Jaunty Jackalope (9.04) seems to hang on the configuration of the network interfaces unless network-manager is installed. This is no longer a problem in Karmic 9.10.

```
apt-get install --no-install-recommends network-
manager
```

Next, you may install more packages as you like, assuming you have the legal rights to redistribute the packages. This is where you build your custom system using packages from the Ubuntu archives.

The customised system can be set-up to allow it to be installed onto machines rather than only ever being a LiveCd. Simply install the Ubiquity packages and an appropriate desktop environment with a window manager. This step is optional and only needed if you want to allow your customised Ubuntu system to be installed on other computers.

For the Gtk front-end

```
apt-get install ubiquity-frontend-gtk
```

For the Qt front-end

```
apt-get install ubiquity-frontend-kde
```

If you installed software, be sure to run

```
rm /var/lib/dbus/machine-id
```

Before exiting the chroot, remove the diversion:

Earlier this guide asked you to make a backup copy of **/sbin/initctl**. If the following command does not restore this file, then restore from the backup copy you made.

```
rm /sbin/initctl
dpkg-divert --rename --remove /sbin/initctl
```

Remove upgraded, old linux-kernels if more than one:

```
ls /boot/vmlinuz-2.6.**-**-generic > list.txt
sum=$(cat list.txt | grep '[^ ]' | wc -l)

if [ $sum -gt 1 ]; then
dpkg -l 'linux-*' | sed '/^ii/!d;/'"$(uname -r |
sed "s/\(.*\)-\([^0-9]\+\)/\1/")"'/d;s/^[^ ]* [^
]* \([^ ]*\).*/\1/;/[0-9]/!d' | xargs sudo apt-
get -y purge
fi

rm list.txt
```

Then just clean up.

```
apt-get clean
```

```
rm -rf /tmp/*

rm /etc/resolv.conf

umount -lf /proc
umount -lf /sys
umount -lf /dev/pts
exit
```

If you also bound your /dev to the chroot, you should unbind that.

```
sudo umount /path/to/chroot/dev
```

So far, you have entered the chroot and installed packages, then cleaned up and left.

There are 4 packages that need to be installed on the Host System which provide the tools to make the Cd image. Syslinux contains isolinux which makes the Cd bootable. Squashfs-tools will compress the image. Genisoimage provides mkisofs tool to turn a directory into a CD image. So install syslinux, squashfs-tools, mkisofs and sbm.

```
sudo apt-get install syslinux squashfs-tools
genisoimage
```

This next command makes the image directory and the 3 required subdirectories.

```
mkdir -p image/{casper,isolinux,install}
# Same as 'mkdir image image/casper
image/isolinux image/install'
```

A. You will need a kernel and an initrd that was built with the Casper scripts. Grab them from your chroot. Use the current version. Note that before 9.10, the initrd was in gz not lz format...

```
cp chroot/boot/vmlinuz-2.6.**-**-generic
image/casper/vmlinuz

cp chroot/boot/initrd.img-2.6.**-**-generic
image/casper/initrd.lz
```

B. If you have a problem with vmlinuz and initrd copying - maybe

you have more than one from these files - you can using following commands:

```
for file in chroot/boot/vmlinuz-2.6.**-**-
generic; do cp $file image/casper/vmlinuz; done

for file in chroot/boot/initrd.img-2.6.**-**-
generic; do cp $file image/casper/initrd.lz; done
```

You need the isolinux and memtest binaries. (Note: some distros place file isolinux.bin under /usr/lib/syslinux .)

```
cp /usr/lib/ISOLINUX/isolinux.bin image/isolinux/
cp /usr/lib/syslinux/modules/bios/ldlinux.c32
image/isolinux/ # for syslinux 5.00 and newer

cp /boot/memtest86+.bin image/install/memtest
```

Boot Instructions for the Remix User

To give some boot-time instructions to the user create an isolinux.txt file in image/isolinux, for example:

```
splash.rle

*****************************************************************

This is an Ubuntu Remix Live CD.

For the default live system, enter "live".  To
run memtest86+, enter "memtest"

*****************************************************************
```

Splash Screen

A graphic can be displayed at boot time, but it is optional. The example text above requires a special character along with the file name of the splash image (splash.rle). To create that character, do the following use the following command:

```
printf "\x18" >emptyfile
```

and then edit the emptyfile with any text editor. Add the file name just next to the first character and add the text you want to display at boot time beneath it and save the file as "isolinux.txt"

To create the splash.rle file, create an image 480 pixels wide. Convert it to 15 colours, indexed (perhaps using GIMP) and "Save As" to change the ending to .bmp which converts the image to a bitmap format. Then install the "netpbm" package and run

```
bmptoppm splash.bmp > splash.ppm

ppmtolss16 '#ffffff=7' < splash.ppm > splash.rle
```

Boot-loader Configuration

Create an isolinux.cfg file in image/isolinux/ to provide configuration settings for the boot-loader. **Please read syslinux.doc** which should be on the host machine in /usr/share/doc/syslinux to find out about the configuration options available on the current set-up. Here is an example of what could be in the file:

```
DEFAULT live
LABEL live
  menu label ^Start or install Ubuntu Remix
  kernel /casper/vmlinuz
  append  file=/cdrom/preseed/ubuntu.seed
boot=casper initrd=/casper/initrd.lz quiet splash
--
LABEL check
  menu label ^Check CD for defects
  kernel /casper/vmlinuz
  append  boot=casper integrity-check
initrd=/casper/initrd.lz quiet splash --
LABEL memtest
  menu label ^Memory test
  kernel /install/memtest
  append -
LABEL hd
  menu label ^Boot from first hard disk
  localboot 0x80
```

```
  append -
DISPLAY isolinux.txt
TIMEOUT 300
PROMPT 1

#prompt flag_val
#
# If flag_val is 0, display the "boot:" prompt
# only if the Shift or Alt key is pressed,
# or Caps Lock or Scroll lock is set (this is the
default).
# If  flag_val is 1, always display the "boot:"
prompt.
#  http://linux.die.net/man/1/syslinux   syslinux
manpage
```

Don't forget to pick the correct extension for your initrd (initrd.gz or initrd.lz). Now the CD should be able to boot, at least it will be after the image is burned

Create manifest:

```
sudo chroot chroot dpkg-query -W
--showformat='${Package} ${Version}\n' | sudo tee
image/casper/filesystem.manifest
sudo cp -v image/casper/filesystem.manifest
image/casper/filesystem.manifest-desktop
REMOVE='ubiquity ubiquity-frontend-gtk ubiquity-
frontend-kde casper lupin-casper live-initramfs
user-setup discover1 xresprobe os-prober
libdebian-installer4'
for i in $REMOVE
do
        sudo sed -i "/${i}/d" image/casper
/filesystem.manifest-desktop
done
```

Compress the chroot

If this Customised Remix is to potentially be installed on some

systems then the /boot folder will be needed. To allow the Customised Cd to be an installer Cd, compress the entire chroot folder with this command:

```
sudo mksquashfs chroot image/casper
/filesystem.squashfs
```

Then write the filesystem.size file, which is needed by the installer:

```
printf $(sudo du -sx --block-size=1 chroot | cut
-f1) > image/casper/filesystem.size
```

However, if it is not going to be installed and is 'only' meant as a LiveCD then the /boot folder can be excluded to save space on your iso image. The live system boots from outside the chroot and so the /boot folder is not used.

```
sudo mksquashfs chroot image/casper
/filesystem.squashfs -e boot
```

It is important to note that if you are building a Karmic LiveCd on an earlier system, you will need the squashfs-tools package from Karmic or the LiveCD will not boot.

Create diskdefines

```
nano image/README.diskdefines
```

example:

```
#define DISKNAME  Ubuntu Remix
#define TYPE  binary
#define TYPEbinary  1
#define ARCH  i386
#define ARCHi386  1
#define DISKNUM  1
#define DISKNUM1  1
#define TOTALNUM  0
#define TOTALNUM0  1
```

Recognition as an Ubuntu Remix

Create an empty file named "ubuntu" and a hidden ".disk" folder. This is needed to make the USB Creator work with this custom iso

image. Without this the image will still boot but the USB creator will not recognize the image as an Ubuntu CD and refuse to use it. Also, create the following files with the pertinent information:

```
touch image/ubuntu

mkdir image/.disk
cd image/.disk
touch base_installable
echo "full_cd/single" > cd_type
echo "Ubuntu Remix 14.04" > info  # Update
version number to match your OS version
echo "http//your-release-notes-url.com" >
release_notes_url
cd ../..
```

Calculate MD5

```
sudo -s
(cd image && find . -type f -print0 | xargs -0
md5sum | grep -v "\./md5sum.txt" > md5sum.txt)
exit
```

This calculates the md5sum of everything in the image folder, except the file named md5sum.txt.

Create iso from the image directory using the command-line

```
cd image
sudo mkisofs -r -V "$IMAGE_NAME" -cache-inodes -J
-l -b isolinux/isolinux.bin -c isolinux/boot.cat
-no-emul-boot -boot-load-size 4 -boot-info-table
-o ../ubuntu-remix.iso .
cd ..
```

The boot.cat file will be automatically created. You may test your image through virtualbox-ose instead of rebooting your real system if you wish.

The USB-Creator works properly with the iso image that has been created so long as the hidden ".disk" folder and its contents are present. If you prefer to do it "by hand", you can put your live

system onto a USB drive yourself. Follow these six steps to do so. You can use these steps to put an existing LiveCd onto a Usb bootable device.

FAT16 file-system (Windows)

1. Prepare your work area:

```
mkdir ../liveusb ../liveusb/mnt
cd ../liveusb
touch loop
```

2. Create a loop device with a fat16 file-system. Use whatever size you need to fit your image; in this case it's a 200Mb sparse file. A sparse file is a file that is bigger than the actual number of bytes it takes up on the disk.

```
dd if=/dev/zero of=loop bs=1 count=1 seek=200M
mkdosfs -F 16 -n rescue loop
```

3. Two options here;

3a Mount the Cd-Rom iso image and your new file-system:

```
mkdir tmp
sudo mount -o loop ../rescue-remix-804Alpha.iso
tmp
sudo mount -o loop loop mnt
```

3b Just use the "image" folder instead of mounting the iso image. This is useful if you don't want to make anything other than a Usb image from scratch (You don't have to make a Cd iso image if you don't need it)

```
ln -s ../image tmp
sudo mount -o loop loop mnt
```

4. Copy the files

```
sudo cp -a tmp/* mnt/
```

5. Change the location of the boot-loader and its configuration file and make it bootable (For fat16 file-system (default))

```
cd mnt
sudo mv isolinux/* .
sudo rmdir isolinux/
```

```
sudo mv isolinux.bin syslinux.bin
sudo mv isolinux.cfg syslinux.cfg
cd ..
sudo umount mnt
sudo umount tmp
syslinux loop
```

6. Pack it up

```
gzip -c loop > remixusb.gz
```

To install onto a usb drive. Insert the drive and identify it's mount-point, for example /dev/sdc. Ensure that the device has a partition table on it and run

```
zcat remixusb.gz | sudo tee /dev/sdc1 >/dev/null
```

Ext2 file-system (proper Linux)

An ext2 file-system is useful in that it can hold larger files and the boot-loader can support relative symlinks. Follow the same steps as above, but substitute the instructions in steps 2 & 5

2. Create an ext2 file-system instead of FAT16, obviously.

```
dd if=/dev/zero of=loop bs=1 count=1 seek=200M
mkfs.ext2 -L rescue -m 0 loop
```

5. It needs to be made bootable *before* unmounting.

```
cd mnt
sudo mkdir boot
sudo mv isolinux boot/extlinux
sudo mv boot/extlinux/isolinux.cfg boot/extlinux
/extlinux.conf
sudo extlinux --install boot/extlinux/
cd ..
sudo umount mnt
sudo umount tmp
```

Partitioning your Usb device

A persistent home can be included within a file instead of a partition. If you want to use a whole partition, do the following.

The Usb image can be installed to any partition on the device. Just make sure that partition is the only one that is marked as bootable. You can partition your Usb storage device to contain the LiveUsb image as well as a storage partition. You can use the storage partition to:

- keep a small amount of recovered files.

- create a persistent home.

- or you can use it as swap space.

To partition your device, see HowtoPartition.

If the storage partition is located after the LiveUsb image partition then Windows won't be able to see it. This is not a problem since you can create the storage partition first and put the live image at the end of the drive. Just make the LiveUsb image partition the only partition flagged as bootable.

When the drive boots the bootable partition will be used and you are good to go. The LiveUsb image's partition won't be seen by Windows.

Troubleshooting

If the device does not boot, you may need to install an MBR onto the device.

```
sudo apt-get install mbr

sudo install-mbr /dev/sdc
```

and try again.

Persistent Data

Create an ext2, ext3 or ext4 partition named "casper-rw" as a separate partition and append the word "persistent" to your "append" line configuration and all your session data will be stored there. You will be able to keep your changes between boots.

```
LABEL live
  menu label ^Start or install Ubuntu
  kernel /casper/vmlinuz
```

```
  append  file=/cdrom/preseed/ubuntu.seed
boot=casper persistent initrd=/casper/initrd.gz
quiet splash --
```

-Update syslinux/isolinux instructions.

With recent versions of syslinux the instructions shown here create a broken menu.

-Graphical boot

http://wiki.debian.org/DebianDesktopMakeSysImageEtch

http://www.sweb.cz/Frantisek.Rysanek/splash/isolinux-splash-HOWTO.html)

If you have any comments or questions, please feel free to add them here.

I wrote some instructions detailing how to create an Ubuntu Live CD from scratch with debian's live-helper scripts: http://david.decotigny.fr/wiki/wakka.php?wiki=LiveHelperUbuntu Tested on an x86_64 jaunty host for an i386 jaunty target CD.

-- DavidDecotigny

Hi, I have a question on this guide. The remix after going through these step does not include Gnome Desktop, right? If so, can someone provide me the package name to the Gnome Desktop?

Thks Alan Chen.

Alan,

The ubuntu-desktop package is based on gnome. If you try to install it with apt-get it will resolve all the dependencies and provide a list. Before installing, it will prompt you and then you can quit. Review the list and you may find the packages you need.

Andrew,

I'd like a live-cd that requires login and does not provide root access without a password of my choosing. Is this possible?

-Patrick

I guess that Casper can do that. Look in the Casper configurations.Perhaps you need to create the user and password

while making the chroot, then tell Casper to only use that user? - Andrew

---

Andrew et al,

I'm not sure that the instructions added here for the issue with running services in the chroot in Karmic are quite right. I say this because I used them today and it resulted in removing the following files: /sbin/start, /sbin/stop, /sbin/initctl. The commands used for removing the diversion did not restore them, and I had to manually reinstall them using a deb file. - Josh

---

-Sergey

I have made all as is written in this instruction. After loading from a disk I get to live cd command line. How I can start installation?

How can I set the keyboard language for the console and X? Currently I can make the livecd without problem but the kbd language is always en. Tnx in advance.

Matteo

People interested in a graphical environment read tasksel documentation.

This command show a list of all live CD/DVD tasks (groups of software)

```
tasksel --list-tasks | grep live
```

and this other the list of packages in the ubuntu-live task

```
tasksel --task-packages ubuntu-live
```

Carles Barreda

Hi, I'm working on a unnattended install CD using Ubuntu Lucid 10.04, and I found that if you erase /sbin/initctl the automatic install doesn't run. Also the instruction to generate the initctl using dpkg-divert is not working. So be careful...

Alejandro

When I follow this guide using 10.04. After the system boots it seems casper wants to run gdm. However, nowhere in this guide

were there instructions for installing gdm and a window manager. Which is exactly what I want. The problem is that casper does not setup any ttys for me. How can I get casper todo this?

-pfifo

How do I install debian-installer is the same as in the alternate-cd, get it to start after instead of the live cd?

Serg

Is it possible to make an HD install from the CD/DVD that has no /pool and /dists folders? I start installation and AptSetup gives me error 127 trying to read the CD pool contents. There's nothing said about that here. As far as I understand this method uses the current (up to date) packages, so we have to grab the cd repo packages from the net repo also if we don't want to break dependencies.

Pry

Pry: If you get an error 127: after install ubiquity open chrooted environment following file:

`/usr/share/ubiquity/plugininstall.py`

and uncomment following line: about 136

`self.configure_apt()`

(put an # sign):

`#self.configure_apt()`

save and test it. If you have an generated LiveCd you can do it in booted Ubuntu Live environment too. Sevoir

---

Yet another question. Is it possible to get the package names to be excluded from the HD install (the difference between filesystem.manifest and filesystem.manifest-desktop files) ? It doesn't seem to be a good idea trying to figure out what to exclude for the next release.

Pry

Successfully completed this great guide! Thanks to everyone who contributed. Now I'm curious, after having followed this, tested out

Reconstructor (meh, not worth $5 in my opinion), and other variants... at what point in this guide could one make some sort of a backup, to start a new build, without having to download the base packages and such? I am assuming after the debootstrap runs? Could I just back up the chroot folder and in the future continue from there with the backup? I'm going to try this, but I want to be certain that there aren't any 'gotchas' with this method.

Derjyn

One possible answer might be another guide in this series

- https://help.ubuntu.com/community/LiveCDCustomization

So, the Cd itself can be considered a back-up stage to be developed from!

Hello,

I just created a bootable/live usb drive with ubuntu 11.04 on it. the problem is that the drive is 16 gigs, but the program I used to create the live usb only allowed me to allocate 4 gigs of extra space (that is space above the space needed for Ubuntu). So I would like to make the extra space much larger perhaps even allow it to fill the whole drive. Is it possible to resize this? and if so how?

Thanks, Daniel Kimbro.

Interesting... I've gone through this procedure a couple of times now. It's awesome! What I don't get though is, something else is happening. If you make changes in /etc/init, those changes won't show up when you boot into the live environment.

I was even more surprised when I found that /home/ubuntu doesn't exist. Assuming this is something Casper does (given that it also takes care of persistent data which gets mounted as /home), where does it store this information?

Regards, Nevyn Hira.

It looks like the **sbm** package existed in natty but has since been eliminated. Does anyone know if a separate package needs to be installed in precise or oneiric? I will forge ahead and report back either way on my results.

Thanks, erixoltan

This howto is excellent! I am, however, having a problem getting my wireless networking credentials onto the custom image. I've tried to do this simply by copying over /etc/NetworkManager/system-connections/* but this is causing problems. When I boot to the image, it says the installer encountered an "unrecoverable error" and then goes to a desktop session without any problem.

The specific error is in /usr/lib/ubiquity/bin/ubiquity:

```
DBusException in call_blocking():
org.freedesktop.NetworkManager.Settings.PermissionDenied:
uid 999 has no permission to perform this
operation
Error: [Errno 13] Permission denied: '/proc
/4512/environ'
```

Any help would be appreciated. Thanks!

Mark

Hi Mark, I think it's a permissions error. /etc/NetworkManager /system-connections need to be owned by root (chown root:root /etc/NetworkManager/system-connections/*) AND only readable and writable by the owner (chmod 600 /etc/NetworkManager /system-connections/*). - Regards, Nevyn.

This howto is great. For those wanting to match the iso release with their local release but cant remember what it is you can run the following one liner to put it in the RELEASE variable

```
RELEASE=`echo $(lsb_release -a 2>/dev/null | grep
Codename)| sed -r 's/^.{10}//'`
echo "Your RELEASE is: "$RELEASE
```

Also please note the change from discover1 to discover in the "remove" line.

- Regards, MrPurple

I cannot `umount chroot/dev` after the chroot process. It says that it is busy. It worked when I used the `-l` flag with umount. Is there something I am missing to do it the way it is outlined here?

Jesse