- Partners
- Support
- Community
- Ubuntu.com

- Page History
- Login to edit

Search

LiveCDCustomization

Contents

1. Why customize LiveCDs
2. How to Customise the Ubuntu Desktop CD
3. System Requirements
4. Install pre-requisities
5. Obtain the base system
    1. Extract the physical DVD .iso contents
    2. Extract the SquashFS system
6. Amending the LiveCD Squash Files System
    1. Prepare and chroot
7. Customizations
    1. Apt
        1. Tasks
    2. DebConf
8. Ending chroot and cleaning up
    1. Before exiting chroot
        1. Cleanup old kernel versions
        2. Cleanup apt and packages
        3. Cleanup temporary files
        4. Terminate / unmount in-chroot bindings and mounts
        5. Exit chroot
    2. After exiting chroot
        1. Remove further mounts and bindings set outside chroot
        2. Remove any temporary package sources from sources.list
        3. Remove bash history from Squash File System
9. Updating and altering the SNAP package configuration
10. Updating the kernel and initrd on the LiveCD proper
    1. Copying the kernel
    2. Extracting the current initrd file
    3. Copying required files to a seed folder
    4. Creating a new initrd / initramfs file
        1. A note on kernel versioning
    5. Clean up the squashfs /boot folder
11. Simple Customisations
    1. Changing GNOME desktop settings
    2. Setting regional defaults
        1. Legacy Boot
            1. Change default language of gfxboot
            2. Change default keyboard
        2. UEFI
    3. Customization limits
    4. Miscellaneous Defaults
    5. Advanced Customizations
        1. Removing the (Casper) Autologin
        2. Boot init
12. Producing the CD image
    1. Assembling the file system
    2. Creating the ISO Image
        1. Retrieving information about the existing image
        2. Assembling the ISO file
    3. Testing the CD
    4. Burning the image to a memory stick or other block device
    5. Burning the image to CD
13. Additional uses for the image
14. Comments

Why customize LiveCDs

You may wish to customise the Ubuntu Desktop LiveCD to:

- make your own Ubuntu distribution
- show off a particular application
- localise to a certain language
- remove software packages
- add software packages
- update software packages
- change system defaults (theme, icons, desktop background, panels, browser homepage, etc)

How to Customise the Ubuntu Desktop CD

This guide is for the Desktop LiveCD. There is another page referring to customisation of the Alternative/Server Install CD. Also, there is a guide on how to create a LiveCD from scratch. This guide is based on customisation from an existing Ubuntu 22.04.1 LTS amd64 LiveCD.

System Requirements

- At least 100GB GB of free disk space
- At least 4GB RAM, much more recommended

- `xorriso`, to compile a new ISO image
- `squashfs-tools`, to unpack and recompile the 'live' filesystem
- `binwalk` for investigating the initramfs file, if updating the boot kernel (optional)
- `casper` files are required for recreating initramfs for updating the boot kernel (optional)
- `live-boot` and `live-boot-initramfs-tools` if upgrading the boot kernel, as needed to recreate initramfs (optional)
- QEMU/KVM, VirtualBox or VMware for testing (optional)

It is strongly recommended that you perform the customisation of a LiveCD from an installed copy of that version of Ubuntu. Some operations in this guide (noted at the relevant point) will likely fail otherwise. For example, to customise from an Ubuntu 22.04.1 LTS amd64 LiveCD, you should work on an installed copy of Ubuntu from that version. It may well be sensible to perform the update from a dedicated installation or a dedicated virtual machine as a number of operations can be disruptive to the host.

The architecture (amd64 or i386) to be stored on the LiveCD should be the same as the architecture used to perform the customization. It is not trivial to customize an amd64 LiveCD using an i386 operating system, for example. Also, the Ubuntu system you are using to create a custom image should be of the same major release version as that of the target system. Otherwise, the LiveCD may not run, and / or you may get stuck at some point during the customisation process.

Install pre-requisities

- Make sure that you have installed the needed tools. This guide assumes you are going to use QEMU KVM virtualisation and complete all optional tasks.

Update the package lists:

```
sudo apt-get update
```

QEMU-KVM and Virtual Machine Manager:

```
sudo apt-get install apparmor apparmor-utils bridge-utils libvirt-clients libvirt-daemon-system libguestfs-tools qemu-kvm virt-manager
```

Strict requirements for customisation:

```
sudo apt-get install binwalk casper genisoimage live-boot live-boot-initramfs-tools squashfs-tools
```

Note that some of these packages also install dependency packages which are needed, such as `xorriso`. If you deviate from this guide, you may need to install some additional required packages manually.

Obtain the base system

- Download an official Desktop image from http://releases.ubuntu.com

- Create a working folder with the image to work from.

```
sudo mkdir ~/custom_live_cd
sudo mv ubuntu-22.04.1-desktop-amd64.iso ~/custom_live_cd
sudo cd ~/custom_live_cd
```

Note: this example refers to Ubuntu 22.04.1 Desktop. Replace it with the name of your iso. It is strongly recommended at this point that on the host / build machine you upgrade all packages at this point:

```
sudo apt-get dist-upgrade
```

Then repeat until all packages are up to date. Following this, clean and autoremove old packages:

```
sudo apt-get autoremove
sudo apt-get clean
```

Extract the physical DVD .iso contents

```
sudo mkdir isomount
sudo mount -o loop ubuntu-22.04.1-desktop-amd64.iso isomount
```

```
sudo mkdir extracted
```

We then copy the files from the mount to a working folder using the rsync command. `rsync` is a fast copy utility, amongst other things, and the -a switch preserves the time, date and permissions on the files. The --exclude switch excludes the compressed squash file system from the copy.

```
sudo rsync --exclude=/casper/filesystem.squashfs -a isomount/ extracted
```

Extract the SquashFS system

We now have a local copy of the DVD file system without the squashfs file, which we will want to work on separately. We extract the squashfs file to a separate local folder:

```
sudo unsquashfs isomount/casper/filesystem.squashfs
```

```
sudo mv squashfs-root edit
```

Amending the LiveCD Squash Files System

Most of the customisations in this guide rely on changing the live Ubuntu instance stored in the Squash File System. This is the basis of what will be installed on any target machine, and this also contains the packages and settings available to the user when the LiveCD is booted.

Amending this file system can be as simple as changing files, but can require package and setting changes that are best done when treating this file system as the running system. To do this, we use a tool called, `chroot`.

Prepare and chroot

`chroot` is an operation that changes the apparent root directory for the current running process and its children. In practice, for present purposes, this lets you work in a copy of a Linux file system you have in a sub-folder and update it, remove and install packages like it was the actual running system using commands like `apt-get`. In fact, in this guide this is almost the only reason to `chroot`. Other changes can just be made by amending files.

Unfortunately, significant functionality will be lost on `chroot`, including network functionality which is likely to be necessary for tasks such as updating and installing packages. You can avoid this issue by just binding /run instead before the `chroot`, which will pull your host's resolvconf info into the chroot:

```
sudo mount -o bind /run/ edit/run
```

Depending on your configuration, you *may* also need to copy the **hosts** file

```
sudo cp /etc/hosts edit/etc/
```

The following bindings are also necessary. Before `chroot`:

```
sudo mount --bind /dev/ edit/dev
```

Then `chroot`:

```
sudo chroot edit
```

Then bind from within `chroot`:

```
mount -t proc none /proc
mount -t sysfs none /sys
mount -t devpts none /dev/pts
```

The above commands will mount important directories of your host system to the edit directory. If you later decide to exit from chroot, make sure to unmount all of them before doing so. (See the cleanup chapter below). Otherwise, your host system will become unusable until you reboot it.

Customizations

Apt

The main reason for `chroot`ing into the edit folder is to install, update and remove `apt` packages, especially for updating the kernel version.

Tasks

To view installed packages by size:

```
dpkg-query -W --showformat='${Installed-Size}\t${Package}\n' | sort -nr | less
```

Installing packages:

```
sudo apt-get update
```

```
sudo apt-get install <package1> [<package2>]
```

When removing packages, use purge:

```
apt-get purge package-name
```

It is likely desirable to update all packages to the latest version, including the kernel. If it is desired to update to the latest kernel, both the build machine and the edited environment must have exactly the same kernel version to enable regeneration of the initrd file. If this is not possible, updating the kernel is not recommended. The commands are:

```
sudo apt-get update
```

```
sudo apt-get dist-upgrade
```

Note - if the kernel is updated you will need to follow the instructions for a kernel update in the separate section below, otherwise you may have obsolete files bloating the image, and boot may fail.

The number of package sources available on a LiveCD is very limited. It may therefore be necessary to temporarily edit the `sources.list` file in the `edit` file system to add sources for desired packages. This can be done before and after the `chroot`. It is important to remove any temporary package sources after the `chroot` has been done. Edit the sources file like this:

```
sudo gedit edit/etc/apt/sources.list
```

DebConf

A number of important settings, such as default user groups, are stored in the `debconf` database. This can be queried and edited during a `chroot`. `debconf` is not to be confused with `dconf` dealt with below.

To display default user groups:

```
echo "get passwd/user-default-groups" | debconf-communicate
```

To set default user groups:

```
echo "set passwd/user-default-groups adm cdrom dip lpadmin plugdev sambashare debian-tor lxd" | debconf-communicate
```

These groups are used for the creation of the LiveCD user on boot and are therefore available when running the LiveCD. Any other `debconf` key can be queried and edited as above.

Ending chroot and cleaning up

Before exiting chroot

Cleanup old kernel versions

Find a list of all currently installed kernel versions. An easy way to do this is to list the config files in the /boot folder:

```
ls -r /boot/config-*
```

This will provide a list. For each instance make a note of that version number. Then, list all linux packages:

```
dpkg --list | grep -i "linux-"
```

Then purge all the packages with old version numbers. Do not purge packages with the most current kernel number:

```
apt-get purge <packages>
```

For example:

```
apt-get purge linux-headers-5.15.0-43 linux-headers-5.15.0-43-generic linux-image-5.15.0-43-generic linux-modules-5.15.0-43-generic linux-modules-extra-5.15.0-43-generic
```

If you looked into the edit/boot folder before entering `chroot` and updating, you will have noticed that it contained a kernel file `vmlinuz` and an initrd / initramfs file called `initrd.img`. Both are symlinks pointing at numbered versions and both were, on your working machine, broken symlinks. Before updating, there were no actual kernel or initrd files there. After updating, there are.

When you have finished cleaning up all out of date kernels per the instructions above there should be one version of an actual kernel file, and one initrd file in the edit/boot folder. These will be moved and further worked on later in these instructions.

Cleanup apt and packages

Use the following commands.

```
apt-get autoremove
```

```
apt clean
apt-get clean
```

Cleanup temporary files

```
rm -rf /tmp/* ~/.bash_history
```

```
rm /var/lib/dbus/machine-id
rm /sbin/initctl
dpkg-divert --rename --remove /sbin/initctl
```

Terminate / unmount in-chroot bindings and mounts

Unbind the in-`chroot` bindings set earlier:

```
umount /proc
umount /sys
umount /dev/pts
```

Exit chroot

```
exit
```

After exiting chroot

Remove further mounts and bindings set outside chroot

Execute the following commands:

```
sudo umount edit/run
sudo umount edit/dev
```

Remove any temporary package sources from sources.list

Do not forget to remove temporary package sources:

```
sudo gedit edit/etc/apt/sources.list
```

Remove bash history from Squash File System

If you performed your chroot by hand, as opposed to executing a script, then bash will have stored your keystroke history, which should be deleted.

```
sudo rm edit/root/.bash_history
```

Updating and altering the SNAP package configuration

Modern Ubuntu uses multiple package managers. As well as apt, the current version uses snap, which is a new package manager provided by Canonical with the intention of making some complex software easier to install and manage. It is desirable to update the snap packages on the LiveCD to the latest versions and it may also be desirable to add or remove packages. Updating these is complex as snapd will by default retain multiple versions of snaps even when uninstalling.

Unfortunately, snap does not yet have many deployment customisation tools or work in `chroot`. The best and easiest way to do this correctly at present is to use a build machine with the identical version of Ubuntu as the source LiveCD, to completely remove all snap packages and snap itself, then reinstall the desired packages. This ensures there are no old package versions and that only the desired code is installed.

The program that manages the snaps is distributed as an apt / debian package called snapd.

On the LiveCD, dummy / mount point folders exist. In addition there is a seed folder with details for an initial setup of snapd and its snaps. The snap packages are apparently duplicated - but in fact hardlinked.

Updating the snaps means firstly taking a list of all desired snap packages and dependent packages.

```
sudo snap list --all
```

Then, iterate through and disable all snaps. For each snap use the command:

```
sudo snap disable <snap name>
```

Then, iterate through all folders under `/var/snap` and call `umount` on them to unmount any folders that will cause a failure.

```
cd /var/snap
sudo umount <snap name>
cd /
```

Next, iterate through and uninstall all snaps. This may require iterating through all the snaps more than once as some snaps depend on each other and will fail until depending snaps are removed.

```
sudo snap remove <snap name>
```

Once all snaps are removed, also remove the `snapd` package.

```
sudo apt-get -y purge snapd
```

Check that `snapd` is really gone by checking that /var/snap and /var/lib/snapd no longer exist.

If it is not, redo until all snaps and snapd are gone.

Delete all the snap related files in and under /etc/systemd/system/. These should already be gone - this is just an extra step for safety and to avoid junk data in the target image. Do not delete other files. For safety, this means delete files with names that begin with the word, 'snap'.

Then, reinstall snapd:

```
sudo apt-get -y install snapd
```

Then, download each snap desired for the new image -

```
sudo snap download <snapname>
```

There will be a file with a name of the form <snapname>[_number].assert and one with a name of the form <snapname>[_number].snap. Copy the assert file to /var/lib/snapd/seed/assertions/ and then run -

```
sudo snap ack <assertion file name>
```

- to install it. To install the snap file itself, run

```
sudo snap install <snap file name>
```

Write an entry in /var/lib/snapd/seed/seed.yaml for each snap. The file overall begins, 'snaps:' then each entry follows as shown below, beginning with ' -'.

```
snaps:
  -
    name: bare
    channel: stable
    file: bare_5.snap
```

Once the snaps are installed, mask, stop and disable the snapd service:

```
sudo systemctl mask snapd.service
sudo systemctl stop snapd.service
sudo systemctl disable snapd.service
```

Then, rsync archive the /var/lib/snapd folder into the target image edit folder, removing the old one first. This basically copies the entire snap configuration on the build host that has just been created to the LiveCD -

```
sudo rm -R edit/var/lib/snapd
sudo rsync -avP /var/lib/snapd/ edit/var/lib/snapd
```

Hardlink all the files in the snaps folder to the seed/snaps folder. This will save space in the final squashfs image, which will respect hardlinks -

```
sudo ln edit/var/lib/snapd/snaps/<snapfilename> edit/var/lib/snapd/seed/snaps/<snapfilename>
```

Create mount points in the image - remove edit/var/snap from the image and recreate. So, edit/var/snap should contain a folder for each snap name. Each should in turn contain a folder called common, and a folder with the version number of the snap. This name can be copied from /var/snaps. Finally, create a symlink called current to the version number folder. So, there should be two folders and a symlink in the folder for each snap name.

There is an identical set of mount points under edit/snap, which should be the same but without the common folder for each entry. Also, there is a bin folder which again can be copied from /snap/bin and which contains symlinks.

Finally, the registered systemd services in the images need updating. Delete all the snap related files in and under edit/etc/systemd/system/. Copy the snap related unit files in or under /etc/systemd/system/ to edit/etc/systemd/system/, along with symlinks.

Unmask, enable and start the snapd service.

```
sudo systemctl unmask snapd.service
sudo systemctl enable snapd.service
sudo systemctl start snapd.service
```

Updating the kernel and initrd on the LiveCD proper

If you updated the kernel in the edit folder whilst in `chroot` you will need to update the kernel in the LiveCD proper. This will require recreation of the initrd file.

On the physical LiveCD file system, the Linux kernel is stored in the `/casper` subfolder. Casper is software that enables booting on diverse hardware by identifying appropriate drivers.

Copying the kernel

```
sudo rm extracted/casper/vmlinuz
sudo cp edit/boot/vmlinuz extracted/casper/vmlinuz
sudo chmod 644 extracted/casper/vmlinuz
```

This should copy the kernel currently pointed at to the `/casper` folder on the LiveCD. Depending on the version of the `cp` command used an -L switch may be necessary to make it follow the symlink.

```
sudo cp -L edit/boot/vmlinuz extracted/casper/vmlinuz
```

Extracting the current initrd file

Firstly, we need to extract the existing file to obtain required scripts using the `unmkinitramfs` command:

```
sudo mkdir initrdmount
sudo unmkinitramfs -v extracted/casper/initrd initrdmount
```

Copying required files to a seed folder

The following commands extract the necessary files from the existing extracted initrd file tree to enable us to create a new one which will work with the LiveCD:

```
sudo cp -R initrdmount/main/conf conf
sudo mv conf initrdconf
sudo cp -R initrdmount/main/scripts initrdconf/scripts
```

Creating a new initrd / initramfs file

With this done, the `mkinitramfs` command can be used to recreate the initrd file. This will draw materials from the host system, which is why the kernel versions must match precisely and why the `casper`, `live-boot` and `live-boot-initramfs-tools` must be installed in the host because they provide needed drivers:

```
sudo mkinitramfs -d initrdconf -o ninitrd
sudo rm extracted/casper/initrd
sudo mv ninitrd extracted/casper/initrd
```

A note on kernel versioning

This guide assumes that the reader is creating a customised LiveCD from a running installation of the same version of Ubuntu as is being customised. If the LiveCD root or the Squash File System has had the kernel updated (e.g. via chroot) then the initrd file should be created from a host running the same kernel version. Ordinarily, the host will have been updated and rebooted also. However, if for some reason the host has not been rebooted, then it is possibly to manually specify the kernel version. This will not ordinarily be necessary, however for example:

```
sudo mkinitramfs -d initrdconf -o ninitrd <new kernel>
```

Clean up the squashfs /boot folder

As set out above, there are no linux kernel nor initrd / initramfs files in the edit/boot folder until you update. There are only broken symlinks. Now we have successfully put them in the /casper folder on the physical LiveCD file system we can once again break those symlinks:

```
sudo rm edit/boot/initrd.img-*
sudo rm edit/boot/vmlinuz-*
```

Simple Customisations

These are customisations that only require file system changes without the use of `chroot`. Changes that require altering the initrd file or otherwise have their own sections.

== Custom Wallpapers Wallpapers are stored in /usr/share/backgrounds (so in your working environment edit/usr/share/backgrounds). New wallpapers must be copied into this folder. For example -

```
sudo cp Wallpaper/* edit/usr/share/backgrounds
```

Wallpapers are only available as choices in the settings menus if they are registered in an XML file in /usr/share/gnome-background-properties. Any suitable xml file in this folder is picked up. For example my-wallpapers.xml. Either create a new file, or add <wallpaper> elements to an existing one.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wallpapers SYSTEM "gnome-wp-list.dtd">
<wallpapers>
  <wallpaper>
    <name>My Custom Stuff</name>
    <name xml:lang="en_AU">My Custom Stuff</name>
    <name xml:lang="en_GB">My Custom Stuff</name>
    <name xml:lang="en_US">My Custom Stuff</name>
    <filename>/usr/share/backgrounds/my-default.png</filename>
    <options>zoom</options>
    <pcolor>#2c001e</pcolor>
    <scolor>#2c001e</scolor>
    <shade_type>solid</shade_type>
  </wallpaper>
</wallpapers>
```

The default wallpaper is set in the file /usr/share/glib-2.0/schemas/10_ubuntu-settings.gschema.override (so in your working environment edit/usr/share/glib-2.0/schemas/10_ubuntu-settings.gschema.override). The default file is called, 'warty-final-ubuntu.png' and can be replaced simply with sed. The procedure for amending schema defaults is set out below under setting default GNOME desktop settings. For example -

```
sed -i 's/warty-final-ubuntu.png/talis-default.png/g' edit/usr/share/glib-2.0/schemas/10_ubuntu-settings.gschema.override
```

Changing GNOME desktop settings

Many of the desktop settings, including the default favourite applications shown on the dock and the wallpaper are stored in dconf, not to be confused with debconf used above. The dconf default settings are set up by schema files which are compiled to a dconf database by glib-2.0.

The folder /usr/share/glib-2.0/schemas (so in your working environment edit/usr/share/glib-2.0/schemas) contains the schemas as XML files, also containing default values. There are also 'override' files, which allow altering of values without changing the XML files. To change a default setting, edit the relevant XML file OR create / edit an override file. Then, recompile the default database -

```
glib-compile-schemas .
```

Performing `ls` on the contents of the folder will yield something like this -

```
00_org.gnome.shell.gschema.override
10_gnome-shell.gschema.override
10_gnome-terminal.gschema.override
10_gsettings-desktop-schemas.gschema.override
10_ubuntu-dock.gschema.override
10_ubuntu-settings.gschema.override
10_virt-manager.gschema.override
casper.gschema.override
com.canonical.unity.desktop.gschema.xml
com.canonical.Unity.Lenses.gschema.xml
com.github.libpinyin.ibus-libpinyin.gschema.xml
com.ubuntu.login-screen.gschema.xml
com.ubuntu.notifications.hub.gschema.xml
[... lots of XML files ...]
```

The settings come from the XML files, can be overridden by the `.override` files. The numbers at the start of the override filenames denote priority. Higher numbered filed override lower ones. An XML file such as `org.gnome.shell.gschema.xml` may look like this -

```
<schemalist>
  <schema id="org.gnome.shell" path="/org/gnome/shell/"
          gettext-domain="gnome-shell">
    [... lots of keys ...]
    <key name="favorite-apps" type="as">
      <default>[ 'firefox_firefox.desktop', 'org.gnome.Nautilus.desktop', 'org.gnome.gedit.desktop', 'org.gnome.Terminal.desktop', 'org.gnome.DiskUtility.desktop', 'virt-manager.desktop' ]</default>
      <summary>List of desktop file IDs for favorite applications</summary>
      <description>
        The applications corresponding to these identifiers
        will be displayed in the favorites area.
      </description>
    </key>
    [... lots of keys ...]
  </schema>
</schemalist>
```

The corresponding override such as `10_ubuntu-settings.gschema.override` may look something like this:

```
###################
# global settings #
###################

[org.gnome.shell]
favorite-apps = ['firefox_firefox.desktop', 'org.gnome.Nautilus.desktop', 'org.gnome.gedit.desktop', 'org.gnome.Terminal.desktop', 'org.gnome.DiskUtility.desktop', 'virt-manager.desktop']
```

In order for desired changes to be effective, you will need to identify the setting key you want to change, find the XML file for it and any override files, and create a higher numbered override to change it. If there is no existing override, you could also change the XML, or change both anyway.

After editing the schema and / or keyfiles, delete and then recreate them using `glib-compile-schemas`. For example whilst in the folder -

```
sudo rm gschemas.compiled
sudo glib-compile-schemas .
```

This operation is file / folder based, meaning as long as the same version of GLIB is on the host machine, there is no need to `chroot`.

Setting regional defaults

Legacy Boot

Change default language of gfxboot

This customization must be done outside the chroot.

```
sudo apt install dpkg-dev uck
apt source gfxboot-theme-ubuntu gfxboot
cd gfxboot-theme-ubuntu*/
make DEFAULT_LANG=fi
sudo cp -af boot/* ../extract-cd/isolinux/
```

Change "fi" to your preferred locale. Note that this does not change which languages are available in the F2 menu. For more info about gfxboot customization, see Ubuntu Customization Kit.

- A other way to change the default language of gfxboot without rebuild the packages is to create a file name **lang** in the isolinux directory containing your locale's name.

Change default keyboard

In 12.04, changing the default keyboard for a live boot is not as obvious as it used to be in 10.04 and earlier. As an example, if you want to keep the default language as English but set the default keyboard to "GB", you are going to have to edit the file /usr/lib/ubiquity/ubiquity/misc.py. If you look at the hard-coded defaults table at around line 620 it should be fairly obvious what's going on...

```
// In /usr/lib/ubiquity/ubiquity/misc.py line 620:
    default_keymap = {
        'ar': 'ara',
        'bs': 'ba',
        'de': 'de',
        'el': 'gr',
        'en': 'us',
etc...
```

You can se the default keymap for your default language by editing this table, but remember that if you update the ubiquity package on your image then any changes will be clobbered.

Also note this only works for X. To set the console keyboard:

```
dpkg-reconfigure keyboard-configuration
```

UEFI

UEFI booting uses GRUB instead of gfxboot. There's no options for changing the language or keyboard for the live session, so it's all English by default. Therefore GRUB configuration needs to be modified. In the Ubuntu Finnish Remix script this is done as follows:

```
sed -i '6i    loadfont /boot/grub/fonts/unicode.pf2' boot/grub/grub.cfg
sed -i '7i    set locale_dir=$prefix/locale' boot/grub/grub.cfg
sed -i '8i    set lang=fi_FI' boot/grub/grub.cfg
sed -i '9i    insmod gettext' boot/grub/grub.cfg
sed -i 's%splash%splash locale=fi_FI console-setup/layoutcode=fi%'
boot/grub/grub.cfg
sed -i 's/Try Ubuntu without installing/Kokeile Ubuntua asentamatta/'
boot/grub/grub.cfg
sed -i 's/Install Ubuntu/Asenna Ubuntu/' boot/grub/grub.cfg
sed -i 's/OEM install (for manufacturers)/OEM-asennus
(laitevalmistajille)/' boot/grub/grub.cfg
sed -i 's/Check disc for defects/Tarkista asennusmedian eheys/'
boot/grub/grub.cfg
mkdir -p boot/grub/locale/
mkdir -p boot/grub/fonts/
cp -a /boot/grub/locale/fi.mo boot/grub/locale/
cp -a /boot/grub/fonts/unicode.pf2 boot/grub/fonts/
```

Customization limits

After customization make sure that there are no users with an UID == 999. Otherwise your image won't boot because no initial user is available (see /usr/share/initramfs-tools/scripts/casper-bottom/25adduser, -> 'db_set passwd/user-uid 999'). This may happen, for example, after installing VirtualBox/GuestAdditions

While in chroot:

```
awk -F: '$3 == 999' /etc/passwd
```

If you get any hits, try changing the uid:

```
usermod -u 500 $hit
```

Miscellaneous Defaults

You may wish to edit the files in /etc/default to change system behavior at startup. You might also edit /etc/profile, /etc/bash.bashrc, and /etc/bash_completion to change login settings for all users on the system. You cannot directly edit defaults for the live cd user (e.g., *casper*, *ubuntu*, or *user*) since that account is created at boot time. You can directly edit root's default files (/root in the chroot environment).

If you wish to change the default timezone used by the live cd, run:

```
dpkg-reconfigure tzdata
```

If you have added a locale and wish to make it the default, update /etc/default/locale. You may have to compile the locale:

```
locale-gen new_locale
update-locale LANG=new_locale LANGUAGE=new_locale LC_ALL=new_locale
```

These changes must be made as root in the chroot environment.

Advanced Customizations

Removing the (Casper) Autologin

The autologin feature of the Jaunty/9.04 live CD is a bit of an on-the-fly boot-hack. After extracting the initrd.gz, you need to edit the casper-bottom/25configure_init script and then recreate the initrd.gz file, replacing the original in extract-cd/casper. The process to do so goes like this:

```
# cd extract-cd/casper
# mkdir tempdir
# cd tempdir
# gunzip -dc ../initrd.gz | cpio -imvd --no-absolute-filenames
# cp scripts/casper-bottom/25configure_init scripts/casper-bottom/25configure_init.orig
# vi scripts/casper-bottom/25configure_init
```

Now look for line 25 which has the conditional statement to test $USERNAME.

Line 25 performs a conditional evaluation and if it evaluates to true, it will execute the code within the if block. The if block contains code to modify files used in the boot process to create the live cd autologin.

To disable the autologin feature, Remove $USERNAME, but just leave the quotes. The -n modifier tests the $USERNAME string to see if it's length is non-zero. By removing the variable, and leaving two double quotes, this statement evaluates to false because the two double quotes effectively make a zero-byte string. Be sure to leave no whitespace between the quotes because whitespace will make the evaluation true and execution wil fall into the if block.

```
21:log_begin_msg "$DESCRIPTION"
22:
23:# Arrange for shells on virtual consoles, rather than login prompts
24:
25:if [ -n "$USERNAME" ]; then
```

After making the change, line 25 will look like this:

```
25:if [ -n "" ]; then
```

Save the file and quit the editor. Then, from extract-cd/casper/tempdir run the following command to re-create the initrd.gz file. There are other methods for re-creating the initrd.gz file on this page which may work also.:

```
# cp ../initrd.gz ../initrd.gz.orig
# find . | cpio -o -H newc | gzip -9 > ../initrd.gz
```

This will create a new initrd.gz file with no auto login. You can then continue to remaster the CD as described on this page. **Be sure to create a user and password to login with before you remaster the cd. If you do not, you will not be able to login after booting!**

Also, I have read a few articles mentioning that Karmic (9.10) uses initrd.lz instead of initrd.gz. I do not know if this is true, but should mention it in case you are not getting the expected results. To unpack the initrd.lz file, you need to do this:

```
# cd extract-cd/casper
# mkdir lztempdir
# cd lztempdir
# lzma -dc -S .lz ../initrd.lz | cpio -imvd --no-absolute-filenames
```

And to re-create the initrd.lz file:

```
# cp ../initrd.lz ../inird.lz.orig
# find . | cpio --quiet --dereference -o -H newc | lzma -7 > ../initrd.lz
```

Boot init

You have to edit the files in edit/usr/share/initramfs-tools/scripts/casper-bottom/* For example you can change the hostname or the livecd user.

i.e.

```
sudo nano edit/usr/share/initramfs-tools/scripts/casper
```

and edit the username or hostname

```
sudo nano edit/usr/share/initramfs-tools/scripts/casper-bottom/10adduser
```

to edit even the livecd user's password.

If you're customizing 10.04, you need to edit variables in /etc/casper.conf for the user and host names instead of modifying the scripts

P.S. in order to obtain an encrypted password, you have to use the mkpasswd program that's shipped with whois package!

Producing the CD image

Assembling the file system

Firstly, regenerate the manifests of the SquashFS file system.

```
sudo chmod +w extracted/casper/filesystem.manifest
sudo chroot edit dpkg-query -W --showformat='${Package} ${Version}\n' > extracted/casper/filesystem.manifest
sudo cp extracted/casper/filesystem.manifest extracted/casper/filesystem.manifest-desktop
sudo sed -i '/ubiquity/d' extracted/casper/filesystem.manifest-desktop
sudo sed -i '/casper/d' extracted/casper/filesystem.manifest-desktop
```

Remove the existing SquashFS file and its signature (this will only apply if you have not followed the instructions, or have already attempted this step before):

```
sudo rm extracted/casper/filesystem.squashfs
sudo rm extracted/casper/filesystem.squashfs.gpg
```

Recreate the SquashFS file from your edited filesystem:

```
sudo mksquashfs edit extracted/casper/filesystem.squashfs -comp xz
sudo printf $(du -sx --block-size=1 edit | cut -f1) > extracted/casper/filesystem.size
```

It is desirable to sign the SquashFS file using your signing key, for example as below (replace the example signing key fingerprint with your own):

```
sudo gpg --local-user 2551D59A01F3022FAFB75644F440B26DF14188A2 --output extracted/casper/filesystem.squashfs.gpg --detach-sign extracted/casper/filesystem.squashfs
```

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

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

Update the image information in extracted/README.diskdefines. An example is below -

```
#define DISKNAME  MyDistribution 1.1 Ubuntu Remix amd64
#define TYPE  binary
#define TYPEbinary  1
#define ARCH  amd64
#define ARCHamd64  1
#define ARCHi386  0
#define DISKNUM  1
#define DISKNUM1  1
#define TOTALNUM  0
#define TOTALNUM0  1
```

Remove the old md5sum.txt file and calculate new md5 sums, as follows:

```
cd extracted
sudo rm md5sum.txt
sudo find -type f -print0 | xargs -0 md5sum | grep -v isolinux/boot.cat | tee md5sum.txt
cd ..
```

Creating the ISO Image

Retrieving information about the existing image

It is necessary before building the image to extract some information and files from the source image being customised. What is needed is the boot sector image, the UEFI boot filesystem and structure information. In modern Ubuntu, the MBR boot code and the EFI image are not provided as files, but they can be extracted from the existing LiveCD image. One is simply in the boot sector of the image, the other is a partition.

Firstly, the source image can be analysed to obtain sector size and partitions using the command -

```
sudo fdisk -l <image name>
```

For example -

```
sudo fdisk -l ubuntu-22.04.1-desktop-amd64.iso
```

This will yield significant information including the offset and size of the EFI partition. For example -

```
ubuntu-22.04.1-desktop-amd64.iso1      64 5465119 7465056  4.1M Microsoft basic
ubuntu-22.04.1-desktop-amd64.iso2 7465120 7473615    8496  4.1M EFI System
ubuntu-22.04.1-desktop-amd64.iso3 7473616 7474215     600  4.1M Microsoft basic
```

Then, extract the MBR, as follows -

```
sudo dd bs=1 count=446 if=<image name> of=mbr.img
```

For example -

```
sudo dd bs=1 count=446 if=ubuntu-22.04.1-desktop-amd64.iso of=mbr.img
```

Then, extract the EFI image, as follows -

```
sudo dd bs=<sector size> count=<sectors in EFI partition> skip=<EFI partition offset> if=<input ISO file> of=EFI.img
```

For example the following, which will extract the whole EFI partition from the `fdisk -l` example output above into a file -

```
sudo dd bs=512 count=8496 skip=7465120 if=ubuntu-22.04.1-desktop-amd64.iso of=EFI.img
```

Now all the components are in place to create a new ISO image. The next thing to do is to create the image using the `xorriso` tool. In previous versions of this guide, the older `mkisofs` tool was used, or `xorriso` running in emulation mode. In this version, native `xorriso` is used and explained.

Firstly, `xorriso` can be used to analyse an existing image and output an example command that is similar to the one used to create the existing image being customised, and has all the necessary switches. The command is as follows:

```
sudo xorriso -indev "<image name>" -report_el_torito cmd
```

For example -

```
sudo xorriso -indev "ubuntu-22.04.1-desktop-amd64.iso" -report_el_torito cmd
```

Assembling the ISO file

The commands shown above will output a list of switches to be used with the `xorriso` command when creating the new ISO image. By default, the command will derive the MBR and EFI images necessary to create the new ISO image, from the source ISO file. As an alternative, an example command is given below with tweaks to use separate MBR and EFI image files, which has the advantage of not needing to refer to the original ISO file. All that is then necessary is to tweak slightly to use the new filesystem as the source, and to use the MBR and EFI image files extracted from the image being customised. For example:

```
sudo xorriso -outdev MyDistribution.iso -map extracted / -- -volid "MyDistribution 1.1 Ubuntu Remix amd64" -boot_image grub grub2_mbr=mbr.img -boot_image any partition_table=on -boot_image any partition_cyl_align=off -boot_image any partition_offset=16 -boot_image any mbr_force_bootable=on -append_partition 2 28732ac11ff8d211ba4b00a0c93ec93b EFI.img -boot_image any appended_part_as=gpt -boot_image any iso_mbr_part_type=a2a0d0ebe5b93344
```

This will yield what should be a bootable ISO file. The next step, which is desirable, is to test. Then hashing and signing the image is also desirable, for example as below (replacing the public key fingerprint with your own):

```
sha512sum MyDistribution.iso >> MyDistribution.iso.sha512
gpg --local-user 2551D59A01F3022FAFB75644F440B26DF14188A2 --output MyDistribution.iso.sha512.gpg --detach-sign MyDistribution.iso.sha512
```

The ISO should boot even in secure boot because the signed Ubuntu shim is still used. However, if unsigned drivers or suchlike are added, then it may not boot in secure boot mode.

Testing the CD

Using the virtual machine environment of your choice (for example QEMU-KVM with Virtual Machine Manager) or other software such as VirtualBox, create a new bare metal virtual machine. Then mount, and boot the ISO. Test with BIOS, UEFI and UEFI with secure boot. Test an installation.

The ISO should boot even in secure boot because the signed Ubuntu shim is still used. However, if unsigned drivers or suchlike are added, then it may not boot in secure boot mode.

If it boots in all configurations, the image is ready to be burned.

Burning the image to a memory stick or other block device

Insert the memory stick and note its device id. (Note, not partition id). Then use the `dd` command:

```
sudo dd if=<input file> of=<output device>
```

For example -

```
sudo dd id=MyDistribution.iso of=/dev/sda
```

Ensure the device has sufficient space for the image. Also ensure that you have the correct device id before doing this as all other information on the target device will be destroyed, or at least rendered hard to retrieve without data recovery software.

Burning the image to CD

Simple! Just do

```
cdrecord dev=/dev/cdrom MyDistribution.iso
```

Additional uses for the image

Install Ubuntu from a USB stick

Installation From Image Loaded On Hard Drive

Comments

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

If you are answering a question, please rewrite the question into a tip that answers the question. (to help keep things to the point.)

New questions at the bottom (I guess.)

I have created an small Customization Example (named Firebird Live CD) by adding an firebird2.1-superand flamerobin packages (this apply to ubuntu Hardy Heron also it was tested with xubuntu 8.04) http://flamerobin.blogspot.com/2008/08/creating-flamerobinfirebird-livecd-with.html

Warning: qemu did not work for me as given in the guide above. Even the normal 8.04 live cd would not boot correctly. Every time, I would get thrown into the ash shell (busybox, initramfs) and while there, a "cat /casper.log" would reveal that it was "Unable to find a medium containing a live filesystem". Just use virtualbox-ose. It actually works with virtualbox. However, after using apt to install virtualbox-ose, I had to run "sudo depmod" again in order for the vboxdrv module to be found by modprobe. Hope that helps!

-rocketman768

Warning: Squashfs is currently in development and is thus not finalized as a format. This means you cannot assume a filesystem.squashfs created using the Ubuntu 9.04 version of makesquashfs will be compatible with the squashfs drive an older live CD. I was customizing an Ubuntu 7.10 LiveCD and when testing it always booted it an (initramfs) prompt--the squashfs was not getting mounted as /. I had to build from within an Ubuntu 7.10 chroot to get it to work.

--Bob/Paul

I have created an small Customization Example (named Firebird Live CD) by adding an firebird-super-server and flamerobin packages (this apply to ubuntu dapper drake) http://flamerobin.blogspot.com/2006/05/creating-flamerobinfirebird-live-cd.html I created an updated guide with Ubuntu Festy Fawn also with an iso download for the Firebird/Flamerobin live cd http://flamerobin.blogspot.com/2007/09/creating-flamerobinfirebird-livecd-with.html

I have created tool for automatic remastering of live CD images. See http://uck.sourceforge.net/ .

Features:

- GUI for simple creation of localized CDs (including changing gfxboot and installing language packs)
- Script for customization of ISO, SquashFS and initrd on live CD.

http://www.atworkonline.it/~bibe/ubuntu/custom-livecd.htm seems to have some nice info. no license that I can see so we would need to ask permission from the author to us its material.

If you want to make the CD boot faster, you might try sorting the files so that they are in the CD in the order that they are accessed: http://lichota.net/%7Ekrzysiek/projects/kubuntu/dapper-livecd-optimization/

Great How To. I am having one issue however. I would like to use custom xorg.conf and sources.list files. Any tips on doing this? Thanks.

- Simply, copy the files to edit/etc/ in the same way (and at the same time) that you copy in the resolv.conf and hosts files.
- I have found that copying xorg.conf doesn't work, as the boot-time scripts overwrite it. Besides, you can't guarantee that a particular xorg.conf will run on all hosts. I'm currently trying to get the binary NVIDIA drivers to work out of the box if an NVIDIA card is present. If I figure out how to fix the xorg.conf, I'll post it here. --JeremyVisser

I've managed to get Synaptic running from within the chroot environment, but it does hang when I try to apply packages. What you do is run "Xnest -ac :1" to get an Xnest server to run on display :1 without access control so anyone can connect to it. Then, in the chroot environment, run "export DISPLAY=:1" to get programs to use the display. Then, type "metacity &" to be able to move windows. Finally, run "synaptic".

It works fine until you try to apply packages, where it hangs for me. --JeremyVisser

Shouldn't the mkinitramfs command use the casper scripts, like "mkinitramfs -o initrd.gz 2.6.15-23-386 -d /usr/share/initramfs-tools"?

There are tricks on how you can get to feel the GNOME system in your chroot environment.

1. Copy your xorg.conf in the chrooted "etc/X11/" directory.

```
cp /etc/X11/xorg.conf edit/etc/X11/
```

2. Create generic devices on your chroot system using MAKEDEV

```
cd /dev/
MAKEDEV generic
```

3. Start X or restart gdm

```
/etc/init.d/gdm start
```

Supposed you want to make modifications on the Desktop, that will be used by all the new users, just change your $HOME to /etc/skel/ and start gdm or X.

```
export HOME=/etc/skel/
```

If you want to load all the other stuff GNOME needs (i.e, dbus, avahi, network-manager), just boot as (single-user mode), and start dbus in your chrooted environment.

```
/etc/init.d/dbus start
```

An example of the whole procedure. (under single-user mode)

```
sudo /etc/init.d/networking restart
sudo nano /run/resolvconf/resolv.conf # See notes about resolv.conf earlier this page. Add temporarily eg. nameserver 8.8.8.8, but do not touch /etc/resolv.conf in any way
sudo cp /etc/X11/xorg.conf edit/etc/X11/
sudo chroot edit
mount -o none /proc
mount -o none /sys
export HOME=/etc/skel/
cd /dev/
MAKEDEV generic
/etc/init.d/dbus start
/etc/init.d/gdm start
```

--- joelbryan

Hello,

I am about to build a new Livecd and have a question: When I change the username, hostname and the user's password the user login automatically during booting the livesystem. But this is not desired. Is it correct, that I have to enter a password under a Desktop LiveCD when I delet the encrypted password in /usr/share/initramfs-tools/scripts/casper-bottom/10adduser ?

Thanks

Changing username and password will not change login behaviour, because this is done in /usr/share/initramfs-tools/scripts/casper-bottom/15autologin

--- Alexander Hosfeld

Having trouble starting a MySQL server from within the chroot jail. Any suggestions?

- Dave

Hey,

I am trying to make a customized live cd of Kubuntu 6.06 that will be completly preconfigured, so that when the user clicks on the install icon on KDE the installer should do everything by itself, meaning that the installer should not ask any questions to the user.

To do this I am trying to write a preseed file to tell the installer the information that it needs.

The problem is that even with this preseed file I could only tell the installer what is the username that it should use, the rest of the information is simply ignored by the installer.

I must be doing something wrong and would appreciate any and all help given me.

Thanks, Komyg

PS: Should I post my preseed file here?

If the livecd is not going to be used for the purposes of installing what files can be removed? Can the "pool" files containing the .debs be removed too?

- Mike

If I want to put in my LiveCD applications that aren't in the sources.list (like ooffice 2.2 or perl audio converter), what cain I do?

- Isoldanne

When you're in the chroot you can install applications just like you would on a live system. If you install from source you can feel free to delete the source tarbal and make folders after you do 'make install'.

- Bob/Paul

To get desired `/etc/X11/xorg.conf` one can modify the `/usr/bin/dexconf`. This script generates xorg.conf automatically in liveCD session according to the booted machine.

For example:

```
# diff ~/bin/dexconf /usr/bin/dexconf
268,269c268
<       Option          "XkbLayout"     "us,il"
<       Option          "XkbOptions" "grp:alt_shift_toggle,grp_led:scroll"
---
>       Option          "XkbLayout"     "$XKB_LAYOUT"
```

-- yotam

I cant do chroot. When i run it a get this error

chroot: cannot run command `/bin/bash': No such file or directory

Please help me out

> I had the same problem because the filesystem that I was working on was FAT32 instead of EXT2 or EXT3. Because of that, /bin/bash was not an executable I guess. If that is your case also, then please try again on a partition that is EXT2 or EXT3 (the Linux type of partition). Good luck, --vvim

Hi, I've created a simple script to ease remastering the Kubuntu Live CD. It uses aufs to avoid copying all the files back and forth.

Maybe it will be usefull to others too. The script must be run as root.

```
CD="${1:-kubuntu-9.04-desktop-i386.iso}" ; shift
# exit after any error:
set -e

which mkisofs mksquashfs tempfile sed

WDIR=`mktemp -d $PWD/kubuntu-remastered.XXXXXXXXXX`
ISO="$WDIR/${CD##*/}"
ISO="${ISO%.iso}-remastered-KDM.iso"
EXIT=""
function addExit {
    EXIT="$@ ; $EXIT"
    trap "$EXIT" EXIT HUP TERM INT QUIT
}
function mnt {
    local margs="$1" ; shift
    local mp="$WDIR/$1"
    for D in "$@" ; do
        mkdir -v -p "$WDIR/$D"
    done
    mount -v $margs "$mp"
    addExit "umount -v $mp"
}

# mount the CD image
mnt "-t auto $CD -o loop,ro" cd

# mount compressed filesystem
mnt "-t squashfs $WDIR/cd/casper/filesystem.squashfs -o ro,loop" sq

# create joined writable filesystem for the new CD
mnt "-t aufs -o br:$WDIR/cd-w=rw:$WDIR/cd=ro none" cd-u cd-w

# create joined writable filesystem for the new compressed squashfs filesystem
mnt "-t aufs -o br:$WDIR/sq-w=rw:$WDIR/sq=ro none" sq-u sq-w

echo ">>> Updating CD content"

(
    cd sq-u

    # DO YOUR CUSTOMIZATION STUFF HERE, CHROOT, MODIFY FILES, ETC.
    # ...
    # ...

)

echo ">>> Compressing filesystem"
mksquashfs $WDIR/sq-u/ $WDIR/cd-u/casper/filesystem.squashfs -noappend

echo ">>> Recomputing MD5 sums"
( cd $WDIR/cd-u && find . -type f -not -name md5sum.txt -not -path '*/isolinux/*' -print0 | xargs -0 -- md5sum > md5sum.txt )

echo ">>> Creating ISO image $ISO"
mkisofs \
    -V "Custom KUbuntu Live CD" \
    -r -cache-inodes -J -l \
    -b isolinux/isolinux.bin \
    -c isolinux/boot.cat \
    -no-emul-boot -boot-load-size 4 -boot-info-table \
    -o "$ISO" \
    $WDIR/cd-u

# The trap ... callbacks will unmount everything.
```

--- Petr Pudlak

> Petr - this looks interesting. How do you use it? When I run it, I get `./fix-live-cd.sh: 1: shift: can't shift that many`. -- Jeff

I need to know how to configure the live cd so that it does not use SWAP!

Please help me!

-- iceman

Hi, I've been experiencing problems with aptitude and I'm pretty sure its because I'm doing all this stuff on NTFS partition (according to this thread http://ubuntuforums.org/showthread.php?t=521905). But I have no other choice,coz' I've got less then 1,7GB on my ext3 partition. Any suggestion ? THX

Error messages I've been getting, when trying to install or update aptitude:

"E: Couldn't make mmap of 25165824 bytes - mmap (19 No such device) W: Unable to munmap E: The package lists or status file could not be parsed or opened."

-heethen (heethen at centrum dot cz)

I would suggest creating an ext3 filesystem within your NTFS partition. To create the file, type

dd if=/dev/zero of=/mnt/hda1/workspace.img bs=1024 count=$((1024*1024*15))

Where I assumed your NTFS partition was on hda1, and you wanted to call the new filesystem workspace.img (it's really just a file within NTFS), and that you wanted 15Gb of room to work. (You don't need that much, but with 15Gb you should have plenty of room.) Then you format the new filesystem with the command

sudo /sbin/mkfs.ext3 /mnt/hda1/workspace.img

then create a mountpoint

sudo mkdir /mnt/workspace

then mount the new filesystem

sudo mount -o loop /mnt/hda1/workspace.img /mnt/workspace

You can then use as much space on the NTFS partition as you want, and have all the benefits of ext3 (like getting the *** thing to work). --Tom

Editing gconf by setting each value separately takes too much time. There is a better way:

Make a test user and adjust the settings as you wish. Run

```
gconftool-2 --dump /the/settings/branch/you/need > ~/live/your-new-settings.xml
sudo chown root:root ~/live/your-new-settings.xml
sudo mv ~/live/your-new-settings.xml ~/live/edit/your-new-settings.xml
```

and then, in the chroot environment, run

```
gconftool-2 --direct --config-source xml:readwrite:/etc/gconf/gconf.xml.defaults --load /your-new-settings.xml
rm /your-new-settings.xml
```

This way you can import the whole branch, e.g. /apps/panel - all settings for Gnome's panels.

---Jacob Popov ubuntu-9.04-desktop-i386.iso

Is there no way to copy the user's gconf-values to the default gconf-values in the chrooted environment? This would be a lot easier than writing commands.

It took me a long time to find the details for configuring a static IP on the live CD without doing it in the root file system. This enables you to create a few boot options with different static details and a DHCP on but all from the same rootfs.

There is a casper parameter which is in the form:

ip=IFACE,ADDRESS,NETMASK,GATEWAY[:IFACE,ADDRESS,NETMASK,GATEWAY]*

--silid

http://ubuntuforums.org/showthread.php?t=839670

I think I have discovered how to do this:

1) Edit accordingly /usr/share/initramfs-tools/scripts/casper-bottom/23networking

2) Run update-initramfs -u -k $version

- Robert

In order to get the livecd to boot from a static address found in the /etc/network/interfaces file.

I edit the /etc/network/interfaces file and add a static address for the interface eth0.

```
auto eth0
iface eth0 inet static
address 10.xxx.xxx.xxx
netmask 255.255.255.0
gateway 10.xxx.xxx.1
```

edit somepath/isolinux/text.cfg add ip=frommedia after splash

```
label live
  menu label ^Try Ubuntu without any change to your computer
  kernel /casper/vmlinuz
  append  file=/cdrom/preseed/ubuntu.seed boot=casper initrd=/casper/initrd.gz quiet splash ip=frommedia --
```

This was a little hard to find. Thanks Robert Nicholas A. Schembi Pittsburgh PA USA

I've tried installing adobe flash player but while it installs fine on chroot, and then shows as installed when booting from CD. Firefox thinks it is not installed and I need to install it manually. Anybody come upon a similar problem

I tried to customize Ubuntu Hardy by installing the RT kernel and did the changes as written in the howto... it goes well, but booting in vbox is really slow. What could be the cause? I tried is once again and also in another vbox installation, but still that problem. Would be nice if someone can point me to an solution! thanks in advance see also thread: http://ubuntuforums.org/showthread.php?p=7334890#post7334890 edit: I think it has something to do with the fact that it is an RT kernel (from Hardy) ... Someone with experience with adding custom RT kernels?

I wrote a script to automate the extraction / chroot / regeneration processes: http://david.decotigny.fr/wiki/wakka.php?wiki=RemasterUbuntu . It has been tested on a jaunty x86_64 host for a jaunty i386 CD image.

-- DavidDecotigny

To put the default user in the "video" group,

```
printf "\nADD_EXTRA_GROUPS=1\nEXTRA_GROUPS="video"\n" >> /etc/adduser.conf
```

-- CarlKarsten

In order to edit a Karmic LiveCD in any release prior to Karmic, it looks like you will need to upgrade squashfs-tools to support SquashFS v4.0:

```
root@maxwell:/home/Z_Custom# unsquashfs mnt/casper/filesystem.squashfs
Filesystem on mnt/casper/filesystem.squashfs is (4:0), which is a later filesystem version than I support!
```

I installed squashfs-tools_4.0-1 from the Karmic release into my Intrepid installation and it seems to be fine now. Maybe we should update these directions to reflect that?

Sincerely,

--John D. Zollo

Hi.. How can I add an a Post Installation script? I mean, I want to execute a script just after the installation process. THis script is simple and set a few things to customize the installation.

Any ideas?

Best Regards

-- Alejandro

Hallo, I'm trying to remaster ubuntu 9.10 livecd. I'm following instruction from this page. When I'm trying to boot from customized system it's seems I can't login to the system, so I can't got the system running.

I am following instruction from limitation part about user id. And make sure there are none of users has uid more then 999. But, after making the squashfs file system I got a message that told me about a user with uid 1000. Actually my host system has a user with uid 1000. I'm using this user to remaster ubuntu livecd.

Does my uid (1000) that I used for remastering the livecd make the new live cd can't login? If it does, how to fix it? Am I must use another user with uid less then 999 to remaster the livecd?

Best Regards

asw_te

Hello, what is the best way to remaster the Karmic release LiveCD (in VirtualBox) to include all the latest updates (more than 150) including new kernel and udev? The kernel changed from 2.6.31-14 to 2.6.31-16 and also udev.

`apt autoremove` removes the old headers, but I have to purge the old kernel manually (`apt purge linux-image-2.6.31-14-generic`).

The first problem is, `update-initramfs -u -k all` still runs for the old kernel, too. Only `aptitude reinstall udev` seems to stop this (`depmod -a && dpkg --configure -a && dpkg-reconfigure udev` does not help).

The second problem is, how do I prepare initrd and isolinux to run and install from a subdirectory (to create a multi-boot DVD)? The base of the DVD is Ubuntu, so / contains *ubuntu-9.10-desktop-i386.iso*, /kubuntu contains *kubuntu-9.10-desktop-i386.iso* and so on. Most things run fine when I introduce the prefix to scripts/casper and /isolinux/text.cfg:

```
LIVE_MEDIA_PATH=casper
$path/.disk/casper-uuid
append  file=/cdrom/preseed/kubuntu.seed boot=casper initrd=/casper/initrd.lz quiet splash --
```

becomes

```
LIVE_MEDIA_PATH=kubuntu/casper
$path/kubuntu/.disk/casper-uuid
append  file=/cdrom/kubuntu/preseed/kubuntu.seed boot=casper initrd=/kubuntu/casper/initrd.lz quiet splash --
```

But the prefixed distributions show only an *Install Ubuntu 9.10* link on the desktop, not Kubuntu (Xubuntu, LUbuntu, ...).

Thanks, René Leonhardt

Hi!

I was hoping to be able to remaster the Ubuntu Live disc so that I can stick it into almost any computer, have it boot up but NOT go into the GUI, just auto login and run a media player with arguements (like a stream address). Was hoping to be able to make my own easy internet radio appliance. Give old pentium computers with soundcards to businesses and they can play my station on their PA for free!

-- Matt

I updated the Karmic Live CD with the latest updates inside chroot using aptitude with update, dist-upgrade, purge linux-image-2.6.31-14-generic, install zsh zsh-doc, install ubuntu-restricted-extras, and enabled DVD playback.

The resulting iso file turned out larger than a CD, and I put it on a USB Disk. The system boots fine, has the latest updates and can play restricted formats 'out of the box'. When I travel, I can just carry a USB Disk instead of a computer!

All releases including Ubuntu 9.10 Karmic Koala now have newer kernels available via updates. The kernel in the live CD ***MUST*** match the ones inside squashfs. If it does not, you get strange problems (like the wireless interface not being detected or CDs not being recognized)

i.e. for Karmic

```
sudo cp edit/boot/vmlinuz-2.6.31-16-generic extract-cd/casper/vmlinuz
sudo cp edit/boot/initrd.img-2.6.31-16-generic extract-cd/casper/initrd.gz
```

Since Karmic uses initrd.lz, I had to uncompress the .gz file and put it back in .lz format.

```
cd extract-cd/casper
sudo mv initrd.lz inird.lz.orig
gzip -dc initrd.gz | sudo lzma -7 > initrd.lz
cd ~/live
```

To conserve space, you can remove the original initrd files before changing the directory back

```
sudo rm -f inird.lz.orig initrd.gz
```

Best, Pradeep Sekar

I have a few questions, I am remastering a copy of 9.04 for a community school program i ran into a few problems

#1 how do you remove the option to update to 9.10 in the update manager #2 how do you get the CD to be recongnized in ubiquity installer as the named distro instead of ubuntu 9.04 (when i go to reinstall the softare still says it is a ubuntu 9.04 cd) #3how do you install a new kernel into the live cd and install part, i was able to upgrade it to 28.18 but when i try putting 31 on it ...it just wont boot.

#4 i have a repos server, i edit the sources.list but ubuntu repos are still the primary, how do i change that over?

#5 finally, i took out gnome games, and i want to add a list of eduacational games, but i want that as a optional install. (some computers won have game son there) is there a way to add the /.Games folder without actaully having data in it? i tried doing this in the chroot adn root menu and it still does not show up on install...

Thank you

ashlessburn

Can anyone tell me how I can take out the option of installing the Live CD at boot. I just want them to be able to use it, not install.

Thanks

--Steve

You can edit the boot options in extract-cd/isolinux/text.cfg Delete all lines and sub-lines of a label you don't like: For example you can delete:

```
label live-install
   menu label ^Install Ubuntu
   kernel /casper/vmlinuz
   append  file=/cdro...
```

-- jancelis

To remove the autologin, you make us work in the extract-cd directory. Does this mean that this process has to be the last one ? Do we have to do it after all the modifications on the edit directory ? Do we have to first create the new user in the chroot ? thx --lsga

I'm having trouble properly adding a repository. How do you add one to a live cd? --Muscovy

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

I customize the live cd and install apache2 , webmin ,,, but how to make it start after the automatic-login ?

Gordon

I use a version of Ubuntu that does not automatically Start the graphical system (startx command) How can I add that to be run at boot?

RJ

I am currently working on a minor update given 10.10 is out. Unfortunately, I got held up and now time has run out. Mainly I wanted to make the scripts more copy/paste friendly. So I added a few env-variables and use those throughout the script. I've done the replacements already, but I didn't get aroud to do some testing.

In the meantime, I'll dump my current version into the above link to save my work.

exhuma

aptitude can't purge packages.

The program 'aptitude' can be found in the following packages:

- aptitude
- aptitude-gtk (You will have to enable component called 'universe')

Try: apt install <selected package>

Unable to find a medium containing a live file system (SOLUTION)

When creating a live-usb with Unetbootin from Windows (Vista/7) you sometimes get this error. A friend of mine suggested another solution than all of the above. Create the USB stick with Unetbootin from a Linux (Ubuntu) box and NOT a windows system. Apparently something is different between the two because now my system just boots flawlessly!

SantanaNL

Regarding the "Unable to find a medium containing a live file system"--

I had this same problem when trying to install Ubuntu 10.04, 10.10, and 10.10 netbook edition on a netbook using a USB stick. I tried creating all three using 10.10's boot disk creator, 11.04's boot disk creator, Unetbootin on 11.04, and Unetbootin on Windows 7. In all cases, I was able to make it to the usb boot menu and would select 'Install'. The Ubuntu logo would show that it was processing for a while and then fail with the above message in a terminal.

In the end, it wasn't a problem with the iso or the created USB stick. The USB drive just wasn't mounted (or would become unmounted after I tell it to go through the 'Install' procedure. Since I couldn't figure out how to boot the usb from the provided terminal, I just tried removing and reinserting the USB drive once I saw the Ubuntu logo and the progress blips.

I was able to get past this problem twice, so far, with this solution.

bananax182

For all of you semi new maybe not quite as knowledgeable Ubuntu users, as myself, the live CD environment does not utilize all of the typical repositories that it would as if you installed it. You might need to add the particular repository of interest to install a program for your live CD.

For instance, I wanted to create a live CD with dcfldd on it. I had to edit etc/apt/sources.list within the chrooted environment to add the universe repository. I uncommented the following to lines in etc/apt/sources.list to install dcfldd:

```
deb http://archive.ubuntu.com/ubuntu natty universe
deb-src http://archive.ubuntu.com/ubuntu natty universe
```

Then just updated apt and installed as normal:

```
apt update
apt install dcfldd
```

Sorry if everyone already knew this. I just figured it may be helpful to users like myself.

mrgrant

Hi 🙂 I'm sure people appreciate comments such as your MrGrant. Otherwise people might feel they are the only ones suffering from something or maybe feel they are to blame.

On a normal LiveCd there is an "installer" icon. DOes anyone know how i can get that icon onto a normal Ubuntu desktop? I have done a full install to a large usb-stick and installed things like "GPartEd" but i definitely miss havign that installer icon & functionality. Regards from Tom 🙂

I am looking to update or see an update to this guide that addresses a custom launcher on unity. So the idea is once the desktop and unity have resolved to the scree, I can click on a custom launcher with out looking in dash to find it. This message should be replaced with this information, in case someone else beats me to the punch.

Chad

I am a total beginner. How do I take my current computer configuration and turn it into a install CD? Thanks!

Nick 🙂 (Sorry if my question sounds dumb)

Hi Nick :)
This link might help

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

but i would avoid it. It's not dumb, just tricky.

Instead you can back-up all your settings, config files/folders and data by backing up the /home folder. You can move your /home to a new partition to separate your data (including configs and stuff) from the Operating System. That allows you to re-install the OS without messing anything up

- https://help.ubuntu.com/community/Partitioning/Home/Moving

It's even possible to have 2 different Operating Systems sharing the same /home although it's not recommended.

Better places to ask questions are

- http://www.linuxquestions.org
- http://www.ubuntuforums.com

and Launchpad
Regards from
Tom 🙂

I think this entry could be improved by adding some explanation of what we are trying to accomplish with these commands. Pretty much any time it says "you need to use" or "remember to" it would be great to know why. Just my two cents.

Mike

"Unable to find a medium containing a live file system" when have customized LiveCD image.

I conveniently missed this step from the main document "Recognition as an Ubuntu Remix": https://help.ubuntu.com/community/LiveCDCustomizationFromScratch#Recognition_as_an_Ubuntu_Remix

As scripts/casper-bottom/41apt_cdrom script from initram image looks for ".disk/info" to recognize the Ubuntu image. If this file does not exist or is zero size on the DVD image, initramfs tools will not mount the CDROM/USB.

```
if [ ! -s /root/cdrom/.disk/info ]; then

    exit 0

fi
```

-Sunil

CategoryDocumentation

LiveCDCustomization (last edited 2024-02-24 22:04:38 by vordrak @ 2a02:c7c:f05e:9100:20bc:4413:298c:f33e[2a02:c7c:f05e:9100:20bc:4413:298c:f33e]:vordrak)