CD-Writing HOWTO

Winfried Trümper <winni@xpilot.org> v2.0, 10 August 1997

This document deals with the process of writing CDs under Linux.

Contents

1 Introduction

My first experience with CD Writers was guided by the "Linux CD Writer mini-HOWTO" by Matt Cutts <cutts@ms.uky.edu>. Thanks Matt!

Although my intention was only to upgrade his document, I rewrote it from the scratch after I realized how much changed since 1994.

1.1 Disclaimer

I (Winfried Truemper) DISCLAIM ALL WARRANTIES WITH REGARD TO THIS DOCUMENT, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A CERTAIN PURPOSE; IN NO EVENT SHALL I BE LIABLE FOR

ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS DOCUMENT.
Short: read and use at your own risk.

1.2 Suggested readings

The *CD-R FAQ* `<http://www.cd-info.com/CDIC/Technology/CD-R/FAQ.html>` is a general FAQ about compact-disk recordables (CD-R).

The *Linux CD-ROM HOWTO* explains everything one should know about CD-ROM drives under Linux. As a supplement, you may want to take a look at the *Linux SCSI HOWTO* and the *Linux Kernel HOWTO*.

1.3 Terminology ... lasers at maximum ... fire!

CD-ROM stands for *Compact Disc Read Only Memory*, a storage medium utilizing an optical laser to sense microscopic pits on a silver shimmering disk. (The silver shimmering comes from an aluminized layer which is the carrier.) The pits represent the bits of the information (in some way) and are so petite that some billions of them fit on the disc. Thus a CD is a mass-storage medium.

The Term *CD-R* is a short form of *CD-ROM recordable* and refers to a CD that doesn't have those ”microscopic pits” on it's surface ... thus it's empty.

Instead of the aluminium layer (silver) it has a special film (colored) into which ”microscopic pits” can be burned in. This is done by giving the laser which normally only senses the pits a little bit more power so he burns the pits. This action can only be taken **once** on a CD-R.

You can leave out some areas for later writing, though, creating a so called multi-session CD.

This mini-HOWTO deals with the task of writing a CD-R. Welcome on board, captain.

1.4 Supported CD-Writers

The detailed list of models which have been reported (not) to work successfully is available from

`<http://www.shop.de/cgi-bin/winni/lsc.pl>`

The list will be included in future versions of this mini-HOWTO.

If your hardware isn't supported you can still use Linux to create an image of the later CD but then you have to use DOS-Software to write the image to the CD-R.
$YoumaywishtodosobecausemostDOS - softwarecannotdealwithlongfilenamesavailableinLinux.$

In this case you can skip all hardware-related sections (those about *generic SCSI devices* and `cdwrite`).

1.5 Supported "features"

Currently the software for burning CDs under Linux does support the following main features:

```
Feature         cdwrite-2.0     cdrecord-1.5
--------------------------------------------
Multisession    only partial    yes

RockRidge       yes (mkisofs)   yes (mkisofs)
El Torito       yes (mkisofs)   yes (mkisofs)
```

RockRidge is an extension to allow longer filenames and a deeper directory hierarchy. *El Torito* can be used to produce bootable CDs. Please see the accompanied documentation for further details upon this special features.

Section 2.8 lists the availability of the mentioned software.

1.6 Mailinglists

If you want to join the development team (with the intention to actively _help_ them), send e-mail to

> cdwrite-request@pixar.com

and put the word `subscribe` in body of the message.

1.7 Availability

The newest version of this document is always available from

> `<http://www.shop.de/ winni/linux/cdr/>`

2 Prepare your Linux-box for writing CD-ROMs

2.1 Set up the hardware

Shut down your computer, switch it off and hook CD writer to the SCSI-bus.

Make sure the SCSI-bus is properly terminated and choose a free SCSI-ID for the writer. Look at the Linux SCSI-HOWTO if you're not sure. If you're completly clueless, ask an expert.

Switch the power on again and check the messages that the BIOS of the SCSI-controller prints immediatly after switching the power on. If it doesn't recognize you writer, go back to step (b). You should see a message like

$missingpicture; hasanyoneascreenshotofit?$

2.2 A note on writing CDs under Linux

In contrast to other rumors, the Linux-kernel does _not_ require a patch in order to write to CDs. Although the file `drivers/scsi/scsi.c` from the kernel-sources contains the lines

```
case TYPE_WORM:
case TYPE_ROM:
 SDpnt->writeable = 0;
```

this does only mean that that CDs and WORMs are not writeable through the standard-devices `/dev/sda`-`/dev/sdh` - which is ok.

Instead of using these devices the writing of CDs is done through the so called *generic SCSI-devices* which permit nearly everything - even writing to CDs.

2.3 Create the generic devices

The *Linux SCSI-HOWTO* says about generic SCSI-devices:

> The Generic SCSI device driver provides an interface for sending SCSI commands to all SCSI devices - disks, tapes, CDROMs, media changer robots, etc.

Speaking of the generic devices as *interfaces* means that they provide an alternate way of accessing SCSI-hardware than through the standard devices.

This alternate way is required because the standard devices are designed to read data block-wise from a disk, tape or cd-rom. Compared to this, driving a cd-writer (or a scanner) is more exotic, e.g commands to position the laser must be transmitted. To have a clean (and therefore fast) implementation of standard-devices, all such exotic actions must be done through the generic scsi devices.

As everthing can be done to SCSI-hardware through the generic devices they are not fixed to a certain purpose - therefore the name *generic*.

Goto the `/dev`-directory and check for generic scsi devices; `ls`-command should show `sga`-`sgh`:

```
bash{$>$} cd /dev
bash{$>$} ls -l sg*
crw-------   1 root   sys     21,   0 Jan  1  1970 sga
crw-------   1 root   sys     21,   1 Jan  1  1970 sgb
crw-------   1 root   sys     21,   2 Jan  1  1970 sgc
crw-------   1 root   sys     21,   3 Jan  1  1970 sgd
crw-------   1 root   sys     21,   4 Jan  1  1970 sge
crw-------   1 root   sys     21,   5 Jan  1  1970 sgf
crw-------   1 root   sys     21,   6 Jan  1  1970 sgg
crw-------   1 root   sys     21,   7 Jan  1  1970 sgh
```

If you don't have those device-files then create them by using the `/dev/MAKEDEV`-script:

```
bash{$>$} cd /dev/
bash{$>$} ./MAKEDEV sg
```

Now the device-files should show up.

2.4 Enable usage of the *generic scsi* and *loopback* devices

The Linux-kernel needs a module that lends it the ability to deal with generic scsi devices. If your running kernel has this feature, it should be listed in the pseudo-file `/proc/devices`:

```
bash{$>$} cat /proc/devices
Character devices:
 1 mem
 2 pty
 3 ttyp
 4 ttyp
 5 cua
 7 vcs
21 sg          {$<$}----- stands for "Scsi Generic device"

30 socksys

Block devices:
 2 fd
 7 loop        {$<$}----- we even can use the loop-devices
 8 sd
11 sr          {$<$}----- stands for "Scsi cd-Rom"
```

Maybe you have to issue the commands `insmod sg`, `insmod loop` or `insmod sr_mod` to load the modules into the kernel. Check again after you've tried this.

If one of them doesn't succeed, you must re-configure your kernel and re-compile it.

```
bash{$>$} cd /usr/src/linux
bash{$>$} make config

[..]
*
* Additional Block Devices
*
Loopback device support (CONFIG_BLK_DEV_LOOP) [M/n/y/?] M

[..]
*
* SCSI support
*
SCSI support (CONFIG_SCSI) [Y/m/n/?]
```

```
*
* SCSI support type (disk, tape, CD-ROM)
*
SCSI disk support (CONFIG_BLK_DEV_SD) [Y/m/n/?] Y
SCSI tape support (CONFIG_CHR_DEV_ST) [M/n/y/?] M
SCSI CD-ROM support (CONFIG_BLK_DEV_SR) [M/n/y/?] M
SCSI generic support (CONFIG_CHR_DEV_SG) [M/n/y/?] M

[..]
ISO9660 cdrom filesystem (CONFIG_ISO9660_FS) [Y/m/n/?] M
```

Please note that I omitted the not-so-important questions.

2.5 Build and install the kernel

If you have questions regarding to this the Linux Kernel-HOWTO is the suggested reading. Furthermore your Linux-distribution should ship with some documentation about this issue.

$Hint: whilere-compiling, youcancontinuewithsteps 2.7-2.9$

2.6 Reboot the computer for the changes to take effect.

Don't panic if the Linux-kernel prints the messages faster than you can read them, at least the initialization of SCSI-devices can be re-displayed with the command `dmesg`:

```
scsi0 : NCR53c{7,8}xx (rel 17)
scsi : 1 host.
scsi0 : target 0 accepting period 100ns offset 8 10.00MHz
scsi0 : setting target 0 to period 100ns offset 8 10.00MHz

  Vendor: FUJITSU   Model: M1606S-512       Rev: 6226
  Type:   Direct-Access                     ANSI SCSI
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0

  Vendor: NEC       Model: CD-ROM DRIVE:84   Rev:  1.0a
  Type:   CD-ROM                            ANSI SCSI
Detected scsi CD-ROM sr0 at scsi0, channel 0, id 4, lun 0

scsi : detected 1 SCSI disk total.
SCSI device sda: hdwr sector= 512 bytes. Sectors= 2131992
```

Shown above is only that part of the initialization-messages that report the detection of physically present SCSI-devices.

2.7 Create *loopback devices*

Goto the `/dev`-directory and check for *loopback devices*. It's not critical if you don't have those devices, but it's convenient if you do (see 3.5). If you already have them, the `ls`-command should show `loop0-loop7`:

```
bash{$>$} cd /dev
bash{$>$} ls -l loop*
brw-rw----   1 root  disk    7,   0 Sep 23 17:15 loop0
brw-rw----   1 root  disk    7,   1 Sep 23 17:15 loop1
brw-rw----   1 root  disk    7,   2 Sep 23 17:15 loop2
brw-rw----   1 root  disk    7,   3 Sep 23 17:15 loop3
brw-rw----   1 root  disk    7,   4 Sep 23 17:15 loop4
brw-rw----   1 root  disk    7,   5 Sep 23 17:15 loop5
brw-rw----   1 root  disk    7,   6 Sep 23 17:15 loop6
brw-rw----   1 root  disk    7,   7 Sep 23 17:15 loop7
```

If you don't have those device-files, then create them by using the `/dev/MAKEDEV`-script:

```
bash{$>$} cd /dev/
bash{$>$} ./MAKEDEV loop
```

The last command only succeeds if you have the `loop`-module in your kernel (see 2.4 for handling of modules). If `insmod loop` does not help, you must wait until the new kernel is properly installed (see 2.5).

2.8 Get the user-software for burning CDs

2.8.1 Command line utilities

The following package is required to generate prototypes of CD-Rs:

<ftp://tsx-11.mit.edu/pub/linux/packages/mkisofs/mkisofs-1.11.tar.gz>

Depending on the model of your cd-writer (see 1.3), one of the following software for writing prototypes to CD-Rs is required:

<ftp://sunsite.unc.edu/pub/Linux/utils/disk-management/cdwrite-2.0.tar.gz>
<ftp://ftp.fokus.gmd.de/pub/unix/cdrecord/cdrecord.tar.gz>

Please use the nearest mirrors of these ftp-Servers or get them from a CD.

Be absolutly sure you have version 2.0 of `cdwrite` or newer. No older version and especially no beta-versions will work properly! Don't trust the man-page of (old) `mkisofs` which states you need version 1.5 of `cdwrite`.

For information about ports of cdwrite to Irix and AIX visit the URL

<http://lidar.ssec.wisc.edu/ forrest/>

The Debian-Distribution ships with a patch for release 1.05 of `mkisofs` that adds the option '-K' to it (see 3.4); it's available from

<ftp://ftp.debian.org/pub/debian/rex/source/misc/mkisofs_1.05-3.diff.gz>

This patch is only necassary if you want to mount the CD-image via the loopback-device (see 3.5.).

A fix for `scsi.c` (kernel) to make this patch superfluous was written by Eric Youngdale and will be included in the next stable release of the Linux kernel.

2.8.2 A graphical user interface (optional)

X-CD-Roast is full X based CD-Writer-Program, and it is the successor of the `cdwtools-0.93`. It's available from

<http://www.fh-muenchen.de/home/ze/rz/services/projects/xcdroast/e_overview.html>

Currently X-CD-Roast is based on a patched version of cdwrite-2.0 and thus comes with exactly the same features (see 1.4). Future versions may be based on the alternate cdrecord software.

3 "If to smoke you turn I shall not cease to fiddle while you burn." $Emperorneroaboutburninghisownclassic - CDs[AD64$; he misunderstand it completly]

Usally the writing of a CD under Linux is done in 2 steps:

- packaging the desired software into one big file using the `mkisofs`-utility
- writing the big file to the CD-R with `cdwrite` or `cdrecord`

It also possible to combine the 2 steps into one via a pipe but that is discouraged because it's not reliable. See below.

3.1 Determine which generic scsi device the writer is attached to

$Pleasenote$: $thecurrentschemefornamingscsidevicesunderLinuxisunnecessarycomplicatedandnotreliableenough.ThefactthatIdescribeitingreaterdetailhereshouldnotbemisinterpretedasaconfirmationofthisscheme.$

After following all steps of the second chapter your system should be able to deal with the task of writing CDs. This section can be used as a proof that everything works as intended.

Issue the command `dmesg`. It should report the messages of the Linux-kernel including those printed while booting (limitation: only the last 200) and contain some information about the CD-Writer connected to the SCSI-bus.

Simple example:

```
  Vendor: YAMAHA  Model: CDR100       Rev: 1.11
  Type:   WORM                        ANSI SCSI revision: 02
Detected scsi CD-ROM sr1 at scsi0, channel 0, id 3, lun 0
```

This machine has 4 SCSI-devices connected to it (you can't see it so I tell you), with SCSI-id's from 0 to 3. The writer is the 4th physically present SCSI-device and therefore connected to `/dev/sgd` (the fourth generic SCSI-device when counting is started with the letter a). In this case the command

```
cdwrite  --eject  --device /dev/sgd
```

opens the tray and is a test if everything is set up properly. A more complicated example:

```
scsi0 : AdvanSys SCSI 1.5: ISA (240 CDB)
scsi1 : Adaptec 1542
scsi : 2 hosts.

  Vendor: HP      Model: C4324/C4325  Rev: 1.20
  Type:   CD-ROM                      ANSI SCSI revision: 02
Detected scsi CD-ROM sr0 at scsi0, channel 0, id 2, lun 0

  Vendor: IBM     Model: DPES-31080   Rev: S31Q
  Type:   Direct-Access               ANSI SCSI revision: 02
Detected scsi disk sda at scsi1, channel 0, id 0, lun 0

scsi : detected 1 SCSI cdrom 1 SCSI disk total.
SCSI device sda: hdwr sector= 512 bytes.
```

In this example two SCSI-controllers host 1 SCSI-device each. What a waste (they are able to host up to 7 devices each). It's not my setup so stop asking if I have too much money ... Anyway for the purpose of being an overlookable example this setup is just excellent. :-)

In the above example the CD-Writer has SCSI-id 2 but it is associated with the first generic SCSI-device `/dev/sga` because it's the first physically present SCSI-device which Linux has detected. Hopefully this shows clearly that the SCSI-id of a device has nothing to do with the associated generic device.

Two questions are left: what happens if you catch the wrong device? If you neither specify the option "–<MANUFACTURER>" nor write any data to the device, usally a warning message is printed and nothing bad happens:

```
bash{$>$} cdwrite  --eject  --device /dev/sgb

Unknown CD-Writer; if this model is compatible with any
```

```
supported type, please use the appropriate command line
flag.

Manufacturer:  IBM
Model:         DPES-31080
Revision:      S31Q
```

In this case the device `/dev/sbg` is a scsi harddisk (from IBM).

If you write data to the wrong device, you overwrite the original content of it and probably irrecoverable damage your system. Be careful, it already happened to me by accident.

3.2 Collect software

Usally this takes up longer than one expects. Remember that missing files cannot be added once the CD is written. :-)

Also keep in mind that a certain amount of the free space of a CD is used for storing the information of the iso9660-filesystem (usally a few MB).

3.3 Storing data on a CD.

The term *iso9660* refers to the format in which data is organised on the CD. To be more precise: it's the filesystem on the CD.

Of course the appearance of files stored in this format is unified by the Linux-kernel as for every other filesystem, too. So if you mount a CD into the directory tree, you cannot distinguish it's files from other files ... beside the fact that they are not writeable ... even not for root. :-) (The mechanism used to unify the appearance of files is called *virtual filesystem*, short *VFS*.)

The features of the iso9660 filesystem are not so rich compared to those of the extended-2 filesystem which is normally used under Linux. On the other hand, the CD is only writable once and some features make no sense anyway. The limitations of the iso9660-filesystem are:

- only 8 levels of sub-directories allowed (counted from the top-level directory of the CD) $useRockRidgeExtensionstoenlargethisnumber$
- maximum length for filenames: 32 characters
- 650 MB capacity

3.4 Create an iso9660 filesystem

Before any storage medium (e.g. floppy disk, harddisk or CD) can be used, it must get a filesystem (DOS speak: get formatted). This filesystem is responsible for organising and incorporating the files that should be stored on the medium.

Well, a writable CD is only writable _once_ so if we would write an empty filesystem to it, it would get formated - but remain completely empty forever. :-)

So what we need is a tool that creates the filesystem while copying the files to the CD. This tool is called `mkisofs`. A sample usage looks as follows:

```
mkisofs  -r -K  -o cd_image   private_collection/
                  `---------'   `-----------------'
                       |                 |
              write output to   take directory as input
```

The option '-r' sets the permissions of all files to be public readable on the CD and enables Rock Ridge extensions. That is what one usually wants and use of this option is recommended until you know what you're doing (hint: without '-r' the mount-point gets the permissions of `private_collection`!). The option '-K' fixes a bug in the Linux kernel and prevents the last file on the CD from being "destroyed" (not really, but Linux can't read it). You need the patched version of mkisofs for it. This option is equivalent to the option '-P' of `cdwrite`. Please see the manual-page of `mkisofs` for details.

`mkisofs` will try to map all filenames to the 8.3-format used by DOS to ensure highest possible compatibility. In case of naming conflicts (different files have the same 8.3-name), numbers are used in the filenames and information about the chosen filename is printed via stderr (usually the screen).

DON'T PANIC:

> Under Linux you will never see these 8.3 filenames because Linux makes use of the Rock Ridge extensions which contain the original file-information (permissions, filename, etc.).

Now you may wonder why the output of `mkisofs` is not directly sent to the writer-device. This has two reasons:

- `mkisofs` knows nothing about driving CD-writers (see section 2.3.)
- It would not be reliable (see section 4.)

Because the timing of the CD-writer is a critical point, we don't feed it directly from `mkisofs` (remember Linux is not a real-time operating system and tasks can be timed badly). Instead it is recommended to store the output of `mkisofs` in a separate file on the harddisk. This file is then an 1:1-image of the later CD and is actually written to the CD with the tool `cdwrite` in a second step.

The 1:1-image gets stored in a huge file so you need the same amount of free disk space that your collected software already eats up. That's a drawback.

One could think of creating an extra partition for that and writing the image to that partition instead to a file. I vote against such a strategy because if you write to the wrong partition (due to a typo), you can lose your complete Linux-system. Furthermore, it's a waste of disk-space because the CD-image is temporary data that can be deleted after writing the CD.

3.5 Test the CD-image

Linux has the ability to mount files as if they were disk-partitions. This feature is useful to check the directory layout of the CD-image is ok. To mount the file `cd_image` created above on the directory `/cdrom`, give the command

```
mount -t iso9660 -o ro,loop=/dev/loop0 cd_image /cdrom
```

Now you can inspect the files under `/cdrom` - they appear exactly as they were on a real CD. To umount the CD-image, just say `umount /cdrom`. Warning: If you did not use the option '-K' for mkisofs then the last file on `/cdrom` may not be fully readable.

Note:

> some ancient versions of `mount` are not able to deal with loopback-devices. If you have such an old version of `mount` it is a hint to upgrade your Linux-system.Several people already suggested to put information about how to get the newest mount-utilities into this mini-HOWTO. I always refuse this. If your Linux-Distribution ships with an ancient `mount`: report it as a bug. If your Linux-Distribution is not easily upgradable: report it as a bug.
>
> If I included all the information that is necessary to work around bugs in bad designed Linux-Distributions, this mini-HOWTO would be a lot bigger and harder to read.

3.6 Remarks on the blank CD-Recordable discs

The german computer magazine "c't" has a list of tips regarding the blank CDs in their november 1996 issue:

- "no-name" discs are generally not of highest quality and should better not be used
- if a recordable CD is defective, this is likely to apply to the whole batch (if you bought more than one at a time); maybe you are lucky and can at least use the first 500MB of such CDs ...
- don't touch the CDs at their shimmering side before writing

3.7 Write the CD-image to a CD

Not much more left to do. Before showing you the last command, let me warn you that CD-writers want to be fed with a constant stream of data because they have only small data-buffers. So the process of writing the CD-image to the CD mustn't be interupted or a corrupt CD will be the result.

To be sure nothing can interupt this process, throw all users of the system and unplug the ethernet-cable ... Read the *Bastard operator from hell* to learn about the right attitude to do so. ;-)

If you are mentally prepared, dress up in a black robe, multiply the SCSI-id of the CD-writer with it's SCSI-revision and light as many candles, speak two verses of the ASR-FAQ and finally type

```
cdwrite  --device /dev/sgd  cd_image
or
cdrecord -v speed=2 dev=4,0 cd_image
```

depending on which software you want to use. Of course you have to replace the example device with the scsi-device your writer is connected to.

Please note that no writer can re-position it's laser and can't continue at the original spot on the CD when it gets disturbed. Therefore any strong vibrations or even a shock will completly destroy the CD you are writing.

3.8 If something goes wrong ...

... remember you can still use corrupt CDs as coasters. :-)

4 Frequently asked questions with answers

4.1 "How sensitive is the burning process?"

Answer: that depends on your writer. Modern ones should have a data-buffer of 1MB or such and can live 1-2 seconds without data. See the manuals or ask your manufacturer if you want to know the details.

Regardless of the size of those data-buffers you must guarantee a constant throughput of 300kb/s or 600kb/s in the long time run.

Disk intensive processes such as updating the *locate*-database lower the maximum flow-rate will surely corrupt the CD; you better check such processes are not started via `cron`, `at` or `anacron` while you burn CD-Rs.

On the other hand, people reported that they compiled a kernel while burning a CD without a glitch. Of course you need a fast machine for such experiments.

4.2 "Has fragmentation a bad impact on the throughput?"

Fragmentation is usally so low that it's impact isn't noticed.

If you're uncertain than look at the messages printed while booting, the percentage of fragmentation is reported while checking the filesystems. You can check for this value with the very dangerous command

```
bash{$>$} e2fsck -n  /dev/sda5       # '-n' is important!
[stuff deleted - ignore any errors]
/dev/sda5: 73/12288 files (12.3% non-contiguous)
```

In this example the fragmentation seems to be very high - but there are only 73 very small files on this filesystem (used as `/tmp`) so the value is _not_ alarming.

4.3 "Is it possible to store the CD-image on an UMSDOS-filesystem?"

Yes. The only filesystem that isn't reliable and fast enough for writing CDs from is the *network filesystem* (*NFS*).

I'm using UMSDOS myself to share the disk-space between Linux and DOS/Win on a PC (486/66) dedicated for writing CDs.

4.4 "Isn't there some way to get around the iso9660 limitations?"

Yes. You can put any filesystem you like on the CD. But other operating systems than Linux won't be able to deal with this CD.

Here goes the recipe:

- Create an empty file of 650MB size.

  ```
  dd if=/dev/zero of="empty_file" bs=1024k count=650
  ```

- Create an extended-2 filesystem on this file

  ```
  bash{$>$} /sbin/mke2fs  empty_file
  empty_file is not a block special device.
  Proceed anyway? (y,n) y
  ```

- Mount this empty file through the loopback-devices

  ```
  mount -t ext2 -o loop=/dev/loop1 empty_file /mnt
  ```

- Copy files to /mnt and umount it afterwards.
- Use `cdwrite` or `cdrecord` on empty_file (which is no longer empty) as if it were an iso9660-image.

If you want to make an entry in `/etc/fstab` for such a CD, disable the checking of it, e.g.:

```
/dev/cdrom  /cdrom  ext2  defaults,ro  0 0
```

The first 0 means "don't include in dumps", the second (=important) one means "don't check for errors on startup" (fsck will fail to check the CD for errors).

4.5 "How to read and write audio CDs?"

Please get the packages "cdda2wav" and "sox", available from sunsite and it's mirrors:

`<ftp://sunsite.unc.edu/pub/Linux/apps/sound/cdrom/cdda2wav0.71.src.tar.gz>`
`<ftp://sunsite.unc.edu/pub/Linux/apps/sound/convert/sox-11gamma-cb2.tar.gz>`

`cdda2wav` enables you to get a specific interval (or a whole track) from your audio CD and converts it into a .wav-file. `sox` converts the wav-files back into the (audio-CD) cdda-format so it can be written to the CD-R using `cdwrite`.

4.6 "How to probe for SCSI-devices after boot?"

The file `drivers/scsi/scsi.c` contains the information

```
/*
 * Usage: echo "scsi add-single-device 0 1 2 3" >/proc/scsi/scsi
 * with  "0 1 2 3" replaced by your "Host Channel Id Lun".
 * Consider this feature BETA.
 *     CAUTION: This is not for hotplugging your peripherals. As
 *     SCSI was not designed for this you could damage your
 *     hardware !
 * However perhaps it is legal to switch on an
 * already connected device. It is perhaps not
 * guaranteed this device doesn't corrupt an ongoing data transfer.
 */
```

4.7 "Is it possible to make a 1:1 copy of a data CD?"

Yes. But you should be aware of the fact that any errors while reading the original (due to dust or scratches) will result in a defective copy.

First case: you have a CD-writer and a seperate CD-ROM drive. By issuing the command

```
cdwrite -v -D /dev/sgc --pad -b  $(isosize  /dev/scd0) /dev/scd0
or
cdrecord -v dev=3,0 speed=2 -isosize /dev/scd0
```

you read the data stream from the CD-ROM drive attached as `/dev/scd0` and write it directly through `/dev/sgc` to the CD-R.

Second case: you don't have a seperate CD-ROM drive. You have to use the writer to read out the CD-ROM in this case:

```
dd if=/dev/scd0 of=cdimage bs=1c count=`isosize  /dev/scd0`
```

This command is equivalent to the result of `mkisofs`, so you should procede as described in chapter 3. Please note that this method will fails on audio CDs!

4.8 "Can Linux read Joliet CDs?"

Yes. But you need to patch the kernel and recompile it. For further details see

<http://www-plateau.cs.berkeley.edu/people/chaffee/joliet.html>

4.9 "How do I read/mount CD-ROMs with the CD-writer?"

Just as you do with regular CD-ROM drives. No tricks at all. Note that you have to use the scd-devices (SCSI CD-ROM) to mount CDs for reading. Example-entry for /etc/fstab:

```
/dev/scd0  /cdrom  iso9660  ro,user,noauto  0  0
```

5 Troubleshooting

5.1 It doesn't work: under Linux

Please check first if the writer works under other operating systems. Concretely:

- Does the controller recognize the writer as an scsi device?
- Does the driver software recognize the writer?
- Is it possible to make a CD using the accompanied software?

If "it doesn't work" even under other operating systems you have a hardware conflict or defective hardware.

5.2 It doesn't work: under DOS and friends

Try to use Linux. Installation and configuration of SCSI-drivers for DOS is the hell. Linux is too complicated? Ha!

5.3 Scsi errors during the burning phase

Most likely those errors are caused by

- missing dis-/reconnect feature on the scsi bus
- unsufficiently cooled hardware
- defektive hardware (should be detected by 5.1.)

Under various circumstances scsi devices dis- and reconnect themselves (electronically) from the scsi bus. If this feature is not available (check controller and kernel parameters) some writers run into trouble during burning or fixating the CD-R.

Especially the NCR 53c7,8xx SCSI driver has the feature disabled by default, so you might want to check it first:

```
NCR53c7,8xx SCSI support                  [N/y/m/?] y
   always negotiate synchronous transfers [N/y/?] (NEW) n
   allow FAST-SCSI [10MHz]                [N/y/?] (NEW) y
   allow DISCONNECT                       [N/y/?] (NEW) y
```

6 Credits

Andreas Erdmann <erdmann@zpr.uni-koeln.de>

provided the example with the YAMAHA-writer

Art Stone <stone@math.ubc.ca>

had the idea to put non-iso9660 filesystems on a CD

Bernhard Gubanka <beg@ipp-garching.mpg.de>

noticed the need of a recent version of mount to utilize the loopback device

Brian H. Toby

polished the wording.

Bruce Perens <bruce@pixar.com>

gave information about the cdwrite-mailinglist

Dale Scheetz <dwarf@polaris.net>

helped improving the section about creating the cdimage.

Edwin H. Kribbs

reported that '-K' requires a patch for mkisofs

Gerald C Snyder <gcsnyd@loop.com>

tested writing of an ext2 CD-ROM (see 4.4)

Ingo Fischenisch <ingo@mi.uni-koeln.de>

provided the example with 2 controllers hosting 2 devices

Janne Himanka <shem@oyt.oulu.fi>

pointer to kernel patch to read Joliet CDs

Joerg Schilling <schilling@fokus.gmd.de>

information about cdrecord

Jos van Geffen <jos@tnj.phys.tue.nl>

noted the problem in 4.9.

Markus Dickebohm <m.dickebohm@uni-koeln.de>

Pierre Pfister <pp@uplift.fr>

helped to develop the recipe on 1:1 copies.

Rick Cochran <rick@msc.cornell.edu>

hint about dis-/reconnect disabled by default in the ncr driver

Stephan Noy <stnoy@mi.uni-koeln.de>

information and experience about writing audio-CDs

Stephen Harris <sweh@mpn.com>

contributed hint about writing audio-CDs

The Sheepy One <kero@escape.com>

suggested using defective CDs as coasters for drinks

Volker Kuhlmann <kuhlmav@elec.canterbury.ac.nz>

noticed that the "cdwrite"-package does not contain `mkisofs`

End of the Linux CD-Writing mini-HOWTO