NAME

xorrecord − Emulation of CD/DVD/BD program cdrecord by program xorriso

SYNOPSIS

xorrecord [options] dev=device [track_source]

DESCRIPTION

xorrecord writes preformatted data to CD, DVD, and BD media.

It understands some options of program cdrecord from cdrtools by Joerg Schilling. Its implementation is part of program xorriso which shares no source code with cdrtools, but rather makes use of libburn for communicating with the drive.
Another, more complete cdrecord emulator is program **cdrskin** which uses the same burn functions as **xorrecord**, but is able to burn audio CDs and to handle CD−TEXT.

MMC, Session, Track, Media types:

MMC is a standard out of the SCSI family which defines the interaction between computers and optical drives. Since more than a decade all CD, DVD, or BD recorders obey this standard regardless by what bus cabling they are attached to the computer. libburn relies on this standard compliance and on the capability of the operating system to perform SCSI transactions over the particular bus cabling.
A **Session** is a data region on an optical disc which usually gets written in a single sweep. It contains at least one **Track** which is a contiguous string of readable blocks. **xorrecord** produces a single session with a single data track which consists of blocks with 2048 bytes each. It chooses the write mode automatically according to media type, medium state, and option −multi.
On CD media there are other track types, like audio, and particular write modes like TAO and SAO. CD and DVD− media can put more than one track into a session. Some of these features can be addressed by program **cdrskin**.
MMC describes several recordable **media types** which roughly form two families.
Sequentially recordable media are CD−R, CD−RW, DVD−R, DVD−R DL, DVD−RW, DVD+R, DVD+R DL, BD−R. Except DVD−R DL they can store more than one session if there is still unwritten space and if the previous session was written with option **−multi**. CD−RW and DVD−RW can be blanked in order to be re−usable from scratch.
Overwritable media are DVD−RAM, DVD+RW, formatted DVD−RW, BD−RE. They offer a single session with a single track for random access writing. There is no need to blank overwritable media before re−use.
DVD−RW media are sold in sequentially recordable state but can be formatted once to become overwritable. See options **blank=format_overwrite** and **blank=deformat**.
If ISO 9660 filesystems are to be stored on overwritable media, then it is possible to emulate multiple sessions, by using option **−−grow_overwriteable_iso**. In this case, the need for blanking before re−use is emulated too.

Drive preparation and addressing:

The drives, CD, DVD, or BD burners, are accessed via file addresses which are specific to libburn and the operating system. Those addresses get listed by a run of **xorrecord −−devices** or **xorriso −device_links**.
On GNU/Linux, FreeBSD, and NetBSD, the user needs rw−permission for the device file. On Solaris, the user needs r−permission and privilege "sys_devices", which is usually gained by running **xorrecord** via command pfexec.
These permissions or privileges are needed already for listing a drive. So it might be necessary to get the overview as superuser or via pfexec.
xorrecord does not perform cdrecord option −scanbus and does not accept the addresses of form Bus,Target,Lun which are told by −scanbus. If support for these addresses is necessary, consider to use program cdrskin.
It is possible to let **xorrecord** work on emulated drives. Their addresses begin by prefix "stdio:" followed by a file address. The emulated media behavior depends on the file type. See man xorriso for details.
If standard output is chosen as emulated drive, then all program result texts, which usually appear on standard output, will get redirected to standard error.

Relation to program xorriso:

xorrecord is actually a command mode of program **xorriso**, which gets entered either by xorriso command "−as cdrecord" or by starting the program by one of the names "xorrecord", "cdrecord", "wodim", or "cdrskin".

This command mode can be left by argument "−−" which leads to generic xorriso command mode. See **man xorriso** for its description. Other than in xorriso command mode, the sequence of the cdrecord emulation options does not matter. All pending actions get performed in a fixed sequence before the program run ends or before cdrecord emulation ends.

OPTIONS

Addressing the drive:

--devices

Print the list of accessible CD, DVD, or BD drives to standard output. Drives might be inaccessible if the user lacks of permissions to use them or if the drive is in use by another program.

Each accessible drive is shown by a line like:

0 −dev ’/dev/sr0’ rwrw−− : ’TSSTcorp’ ’CDDVDW SH−S203B’

The libburn address of this drive is ’/dev/sr0’. ’TSSTcorp’ is the name of the vendor (in this case: Toshiba Samsung Storage Technologies Corporation), ’CDDVDW SH−S203B’ is the model name (in this case: a DVD burner).

Afterwards end emulation without performing any further drive operation.

dev=drive_address

Set the libburn address of the drive to be used.

E.g. on GNU/Linux: dev=/dev/sr0

E.g. on FreeBSD: dev=/dev/cd0

E.g. on NetBSD: dev=/dev/rcd0d

E.g. on Solaris: dev=/dev/rdsk/c2t2d0s2

See also above "Drive preparation and addressing".

The medium in the drive should not be mounted or be otherwise in use.

This option will only get into effect if a track source, a blank= option, or a drive inquiry option is given. Else it will lead to a SORRY event and normally cause a non−zero exit value.

Inquiring drive and media:

−inq Print to standard output: vendor, model name, and firmware revision of the drive.

−checkdrive

Print unconditionally that the drive supports burnfree, SAO, and TAO. Also print the output of option −inq.

−atip Print the output of −checkdrive, the most capable profile of the medium in the drive, the list of profiles which are supported by the drive, whether it is erasable (i.e. can be blanked), the media manufacturer, and the medium product name.

Profiles are usage models, which are often tied to a particular media type (e.g. CD−RW), but may also apply to a family of media. E.g. profile CD−ROM applies to all CD media which contain data.

−toc Print a table of content of the medium in the drive. The output is not compatible to cdrecord option −toc, but rather the one of **xorriso** command −toc. It lists the address, vendor, model name, and firmware revision of the drive.

About the medium it tells product name and manufacturer, whether there is already content written, and if so, whether the medium is closed or appendable. Appendable media can take another session. The amount of readable and writable data is told. If there are sessions, then their start block address and size is reported. If a session contains an ISO 9660 filesystem, then its Volume Id is reported. If the medium is writable, then the next writable block address is reported.

If not option **−−grow_overwriteable_iso** is given or no ISO 9660 file system is present on the medium, then overwritable media are reported as being blank. This is due to the fact that they can be written from scratch without further preparation, and that MMC does not distinguish between

data written by the most previous burn run and older data which have not been overwritten by that burn run. Consequently, these media are reported with 0 readable blocks, although all their writable blocks normally are readable, too.

−msinfo

Print the argument text for option −C of programs mkisofs, genisoimage, or xorrisofs. It consists of two numbers separated by a comma.
The first number tells the first block of the first track of the last recorded session. This is also the address used by default when operating systems mount a medium with e.g. ISO 9660 filesystem.
The second number tells the next writable address, where **xorrecord** will begin to write the next session.
This option is only valid for written, appendable media. In all other cases it will yield no output text but will abort the program with non−zero exit value.

Settings for the burn run:

A burn run requires exactly one track source address argument, which tells from where to read the data which shall be put into the upcoming session. The medium state must be either blank or appendable.
Track source may be "−" for standard input or the address of a readable file of any type except directories. Nearly all media types accept a track source with unpredictable byte count, like standard input or named pipes. Nevertheless, DVD−R DL and DVD−RW blanked by mode deformat_quickest demand exact in−advance reservation of the track size, so that they either need to be read from a source of predictable length, or need to be accompanied by option **tsize=** or by option **−isosize**.
Several options expect a size value as argument. A number with a trailing letter "b" or without a trailing letter is a plain byte count. Other trailing letters cause multiplication of the given number by a scaling factor:
"k" or "K" = 1024 , "m" or "M" = 1024k , "g" or "G" = 1024m , "s" or "S" = 2048
E.g. tsize=234567s means a size of 234567 * 2048 = 480393216 bytes.

blank=mode

Blank a CD−RW or DVD−RW to make it re−usable from scratch. Format a DVD−RW, DVD+RW, DVD−RAM, BD−R, or BD−RE if not yet formatted.
This operation normally makes any recorded data on the medium unreadable. It is combinable with burning in the same run of **xorrecord**, or it may be performed without a track source, leaving the medium empty.
The mode given with blank= selects the particular behavior:

as_needed

Try to make the media ready for writing from scratch. If it needs formatting, then format it. If it is not blank, then try to apply blank=fast. It is a reason to abort if the medium cannot assume thoroughly writeable state, e.g. if it is a non−blank write−once.
This leaves unformatted DVD−RW in unformatted blank state. To format DVD−RW use blank=format_overwrite. Blank unformatted BD−R stay unformatted.
(Note: blank=as_needed is not an original cdrecord option.)

all

Blank an entire CD−RW or an unformatted DVD−RW.

fast

Minimally blank an entire CD−RW or blank an unformatted DVD−RW.

deformat

Like blank=all but with the additional ability to blank overwritable DVD−RW. This will destroy their formatting and make them sequentially recordable.
(Note: blank=deformat is not an original cdrecord options)

deformat_quickest

Like blank=deformat but blanking DVD−RW only minimally. This is faster than full blanking but yields media incapable of writing tracks of unpredictable size. Multi−session will not be possible either.

(Note: blank=deformat_quickest is not an original cdrecord option.)

format_overwrite

Format a DVD−RW to "Restricted Overwrite". The user should bring some patience.
Format unformatted DVD+RW, BD−RE or blank BD−R to their default size. It is not mandatory to do this with DVD+RW and BD−RE media, because they will get formatted automatically on the first write attempt.
BD−R media may be written in unformatted state. This keeps disabled the replacement of bad blocks and enables full nominal write speed. Once BD−R media are written, they cannot be formatted any more.
For re−formatting already formatted media or for formatting with non−default size, use program **xorriso** with command **−format**.
(Note: blank=format_overwrite is not an original cdrecord options)

help

Print a short overview of blank modes to standard error output.
Afterwards end emulation without performing any drive operation.

−multi This option keeps CD, unformatted DVD−R[W], DVD+R, or BD−R appendable after the current session has been written. Without it the disc gets closed and may not be written any more − unless it is a −RW and gets blanked, which causes loss of its content.
This option cannot be applied to DVD−R DL or to DVD−RW which were blanked by mode "deformat_quickest". Option −−multi_if_possible may automatically recognize and handle this situation.
In order to have all filesystem content accessible, the eventual ISO−9660 filesystem of a follow−up session needs to be prepared in a special way by the filesystem formatter program. mkisofs, genisoimage, and xorrisofs expect particular info about the situation which can be retrieved by **xorrecord** option −msinfo.
With overwritable DVD or BD media, −multi cannot mark the end of the session. So when adding a new session, this end has to be determined from the payload. Currently only ISO−9660 filesystems can be used that way. See option **−−grow_overwriteable_iso**.

−dummy

Try to perform the drive operations without actually affecting the inserted media. There is no warranty that this will work with a particular combination of drive and media. Blanking is prevented reliably, though. To avoid inadverted real burning, −dummy refuses burn runs on anything but CD−R[W], DVD−R[W], or emulated stdio−drives.

−waiti Wait until input data is available at stdin or EOF occurs at stdin. Only then begin to access any drives.
One should use this if xorrisofs is working at the end of a pipe where the feeder process reads from the drive before it starts writing its output into xorrisofs. Example:
xorrisofs ... −C 0,12800 −M /dev/sr0 ... | \
xorrecord dev=/dev/sr0 ... −waiti −
This option works even if standard input is not the track source. If no process is piping in, then the Enter key of your terminal will act as trigger for **xorrecord**. Note that this input line will not be consumed by cdrskin if standard input is not the track source. It will end up as shell command, usually.

tsize=size

Announce the exact size of the track source. This is necessary with DVD−R DL media and with quickest blanked DVD−RW, if the size cannot be determined in advance from the track source. E.g. if it is standard input or a named pipe.
If the track source does not deliver the predicted amount of bytes, the remainder of the track is padded with zeros. This is not considered an error. If on the other hand the track source delivers more than the announced bytes then the track on media gets truncated to the predicted size and xorrecord exits with non−zero value.

−isosize

Try to obtain the track size from the content of the track source. This works only if the track source bears an ISO 9660 filesystem. Any other track source content will cause the burn run to abort.

If the track source is not a regular file or block device, then this option will work only if the program's fifo size is at least 64k. See option fs=.

padsize=size

Add the given amount of trailing zeros to the upcoming track. This feature can be disabled by size 0. Default is 300 kB in order to work around a problem with GNU/Linux which often fails to read the last few blocks of a CD track which was written in write mode TAO. TAO is used by **xorrecord** if the track size cannot be predicted or if the CD medium is not blank but appendable.

−nopad

The same as padsize=0.

−pad The same as padsize=15s. This was once sufficient with older GNU/Linux kernels. Meanwhile one should at least use padsize=128k, if not padsize=300k.

−data Explicitly announce that the track source shall be recorded as data track, and not as audio track. This option has no effect with **xorrecord**, because there is no support for other track formats anyway.

−tao Explicitly demand that write type TAO shall be used for CD, or Incremental for DVD−R. Normally the program will choose the write type according to the given medium state, option −multi, and track source. Demanding it explicitly prevents the start of a write run, if it is not appropriate to the situation.

−sao Explicitly demand that write type SAO shall be used for CD, or DAO for DVD−R. This might prevent the write run, if it is not appropriate to the situation.

−dao Alias of −sao.

fs=size Set the size of the program fifo buffer to the given value rather than the default of 4m.

The fifo buffers a temporary surplus of track source data in order to provide the drive with a steady stream during times of temporary lack of track source supply.

Other than cdrecord, xorrecord enables drive buffer underrun protection by default and does not wait with writing until the fifo is full for a first time. On very old CD drives and slow computers, this might cause aborted burn runs. In this case, consider to use program **cdrskin** for CD burning. DVD and BD drives tolerate buffer underrun without problems.

The larger the fifo, the longer periods of poor source supply can be compensated. But a large fifo can deprive the operating system of read cache for better filesystem performance.

speed=value

Set the write speed. Default is 0 = maximum speed. Speed can be given in media type dependent x−speed numbers or as a desired throughput per second in MMC compliant kB (= 1000) or MB (= 1000 kB). Media x−speed factor can be set explicitly by appending "c" for CD, "d" for DVD, "b" for BD. "x" is optional.

Example speeds:

706k = 706kB/s = 4c = 4xCD

5540k = 5540kB/s = 4d = 4xDVD

If there is no hint about the speed unit attached, then the medium in the drive will decide. Default unit is CD, 1x = 176,400 raw bytes/second. With DVD, 1x = 1,385,000 bytes/second. With BD, 1x = 4,495,625 bytes/second.

MMC drives usually activate their own idea of speed and take the speed value given by the burn program only as a hint for their own decision.

minbuf=percentage

Equivalent to:

modesty_on_drive=<percentage>

−immed

Equivalent to:
modesty_on_drive=75
In cdrecord, this also controls use of the Immed bit. But xorriso uses Immed where possible and appropriate, unless it is disabled by option use_immed_bit=off .

−eject Eject the drive tray after alll other work is done.

Program version and verbosity:

−version

Print to standard output a line beginning by
"Cdrecord 2.01−Emulation Copyright"
and further lines which report the version of xorriso and its supporting libraries. They also state the license under which the program is provided, and disclaim any warranty, to the extent permitted by law.
Afterwards end emulation without performing any drive operation.

−v Increase program verbosity by one level. There are four verbosity levels from nearly silent to debugging verbosity. The both highest levels can be enabled by repeated −v or by −vv or by −vvv.

−V Log SCSI commands and drive replies to standard error. This might be of interest if**xorr ecord** and a particular drive or medium do not cooperate as expected, or if you just want to know how libburn interacts with the drive. To understand this extremely verbose log, one needs to read SCSI specs SPC, SBC, and MMC.
Please do not add such a log to a bug report on the first hand, unless you want to point out a particular deviation from said specs, or if you get asked for this log by a maintainer of **xorrecord** who feels in charge for your bug report.

−help Print a sparse list of program options to standard error and declare not to be cdrecord.
Afterwards end emulation without performing any drive operation.

Options not compatible to cdrecord:

--no_rc Only if used as first command line argument this option prevents reading and interpretation of startup files. See section FILES below.

--drive_not_exclusive

This option disables the use of device file locking mechanisms when acquiring the drive. On GNU/Linux the locking is done by open(O_EXCL), on FreeBSD by flock(LOCK_EX).
Be aware that it can cause problems if you use a drive which is mounted, or opened by some other process, or guarded by /dev/pktcdvd*. Make sure that other users of the drive do not cause drive activities while a xorrecord burn run is going on.

drive_scsi_dev_family=sr|scd|sg|default

GNU/Linux specific:
By default, cdrskin tries to map Linux drive addresses to /dev/sr* before they get opened for operating the drive. This coordinates well with other use cases of optical drives, like mount(8). But since year 2010 all /dev/sr* share a global lock which allows only one drive to process an SCSI command while all others have to wait for its completion. This yields awful throughput if more than one drive is writing or reading simultaneously.
The global lock is not applied to device files /dev/sg* and also not with the system calls read(2), write(2). But ioctl(SG_IO) is affected, which is needed to perform the SCSI commands for optical burning.
So for simultaneous burn runs on modern GNU/Linux it is advisable to use drive_scsi_dev_family="sg". The drive addresses may then well be given as /dev/sr* but will nevertheless get used as /dev/sg*.

--grow_overwriteable_iso

Enable emulation of multi−session writing on overwritable media which contain an ISO 9660 filesystem. This emulation is learned from growisofs −M but adapted to the usage model of

xorrecord −msinfo
xorrisofs −C −M | xorrecord −waiti −multi −
for sequential media.
−−grow_overwriteable_iso does not hamper the use of true multi−session media. I.e. it is possible to use the same **xorrecord** options with both kinds of media and to achieve similar results if ISO 9660 filesystem images are to be written. This option implies option −isosize and therefore demands that the track source is a ISO 9660 filesystem image.
With overwritable media and no option blank=fast|all present it expands an eventual ISO 9660 filesystem on media. It is assumed that this image's inner size description points to the end of the valuable data. Overwritable media with a recognizable ISO 9660 size will be regarded as appendable rather than as blank. I.e. options −msinfo and −toc will work. −toc will always show a single session with its size increasing with every added ISO 9660 image.

--multi_if_possible
Apply option −multi if the medium is suitable. Not suitable are DVD−R DL and DVD−RW, which were blanked with mode "deformat_quickest".
Not all drives correctly recognize such fast−blanked DVD−RW which need "on". If there is well founded suspicion that a burn run failed due to −multi, then this causes a re−try without −multi.

stream_recording="on"|"off"|number
Mode "on" requests that compliance to the desired speed setting is preferred over management of write errors. With DVD−RAM and BD this can bring effective write speed near to the nominal write speed of the media. But it will also disable the automatic use of replacement blocks if write errors occur. It might as well be disliked or ignored by the drive.
If a number is given, then error management stays enabled for all byte addresses below that number. Any number below 16s is the same as "off".

dvd_obs="default"|"32k"|"64k"
Linux specific: Set the number of bytes to be transmitted with each write operation to DVD or BD media. Tracks get padded up to the next multiple of this write size. A number of 64 KB may improve throughput with bus systems which show latency problems. The default depends on media type, option stream_recording=, and on compile time options.

modesty_on_drive=parameter[:parameters]
Control whether the drive buffer shall be kept from getting completely filled. Parameter "on" (or "1") keeps the program from trying to write to the burner drive while its buffer is in danger to be filled over a given limit. If this filling is exceeded then the program will wait until the filling reaches a given low percentage value.
This can ease the load on operating system and drive controller and thus help with achieving better input bandwidth if disk and burner are not on independent controllers (like hda and hdb). It may also help with simultaneous burns on different burners with Linux kernels like 3.16, if one has reason not to fix the problem by drive_scsi_dev_family="sg". On the other hand it increases the risk of buffer underflow and thus reduced write speed.
Some burners are not suitable because they report buffer fill with granularity too coarse in size or time, or expect their buffer to be filled to the top before they go to full speed.
Parameters "off" or "0" disable this feature.
The threshold for beginning to wait is given by parameter "max_percent=". Parameter "min_percent=" defines the threshold for resuming transmission. Percentages are permissible in the range of 25 to 100. Numbers in this range without a prepended name are interpreted as "on:min_percent=".
E.g.: modesty_on_drive=75
The optimal values depend on the buffer behavior of the drive.
Parameter "timeout_sec=" defines after which time of unsuccessful waiting the modesty shall be disabled because it does not work.
Parameter "min_usec=" defines the initial sleeping period in microseconds. If the drive buffer appears to be too full for sending more data, the program will wait the given time and inquire the buffer fill state again. If repeated inquiry shows not enough free space, the sleep time will slowly

be increased to what parameter "max_usec=" defines.
Parameters, which are not mentioned with a modesty_on_drive= option, stay unchanged. Default is:
modesty_on_drive=off:min_percent=90:max_percent=95:
timeout_sec=120:min_usec=5000:max_usec=25000

use_immed_bit="on"|"off"|"default"
Control whether several long lasting SCSI commands shall be executed with the Immed bit, which makes the commands end early while the drive operation is still going on. xorriso then inquires progress indication until the drive reports to be ready again. If this feature is turned off, then blanking and formatting will show no progress indication.
It may depend on the operating system whether −use_immed_bit is set to "off" by default.

write_start_address=value
Set the block address on overwritable media where to start writing the track. With DVD+RW, DVD−RAM or BD−RE, byte_offset must be aligned to 2 kiB blocks, but better is 32 kiB on DVD and 64 kiB on BD. With formatted DVD−RW 32 kiB alignment is mandatory.
Other media are not suitable for this option.

stdio_sync="on"|"off"|number
Set the number of bytes after which to force output to emulated stdio: drives. This forcing keeps the memory from being clogged with lots of pending data for slow devices. Default "on" is the same as "16m". Forced output can be disabled by "off".

EXAMPLES

Overview of examples:
Get an overview of drives and their addresses
Get info about a particular drive or loaded media
Prepare CD-RW or DVD-RW for re-use, BD-R for bad block handling
Format DVD-RW to avoid need for blanking before re-use
De-format DVD-RW to make it capable of multi-session again
Write a single ISO 9660 filesystem image
Write multiple ISO 9660 sessions
Write ISO 9660 session on-the-fly
Write compressed afio archive on-the-fly

Get an overview of drives and their addresses:
$ xorrecord −−devices

Get info about a particular drive and loaded media:
$ xorrecord dev=/dev/sr0 −atip −toc −−grow_overwriteable_iso

Prepare CD-RW or DVD-RW for re-use:
$ xorrecord −v dev=/dev/sr0 blank=as_needed −eject

Format DVD-RW to avoid need for blanking before re-use:
$ xorrecord −v dev=/dev/sr0 blank=format_overwrite −eject
This command may also be used to format BD−R media before first use, in order to enable handling of write errors. Several hundred MB of spare blocks will be reserved and write runs on such media will perform with less than half nominal speed.

De-format DVD-RW to make it capable of multi-session again:
$ xorrecord −v dev=/dev/sr0 blank=deformat

Write a single ISO 9660 filesystem image:
$ xorrecord −v dev=/dev/sr0 speed=12 fs=8m \
blank=as_needed −eject padsize=300k my_image.iso

Write multiple ISO 9660 sessions:
This is possible with all media except minimally blanked DVD−RW and DVD−R DL, which cannot do multi−session.

The first session is written like in the previous example, except that option −multi is used. It will contain the files of hard disk directory ./tree1 under the ISO 9660 directory /dir1:

```
$ xorrisofs −o image_1.iso −J −graft−points /dir1=./tree1
$ xorrecord −v dev=/dev/sr0 speed=12 fs=8m \
      −multi −−grow_overwriteable_iso \
      blank=as_needed −eject padsize=300k image_1.iso
```

For the second session xorrisofs needs to know the −msinfo numbers of the medium. Further it will read data from the medium by using the system's read−only CD−ROM driver.

Many systems do not take notice of xorrecord's write activities. It is necessary to force their attention by ejecting and reloading the drive tray. Therefore above run uses option −eject.

Get the −msinfo numbers (and properly reload the tray if it has a motor) by:

```
$ m=$(xorrecord dev=/dev/sr0 −msinfo)
```

Offer a victim to any problem caused by obtrusive demons after tray loading:

```
$ dd if=/dev/sr0 count=1 >/dev/null 2>&1
```

Use the numbers with xorrisofs to add ./tree2 to the image as /dir2:

```
$ xorrisofs −M /dev/sr0 −C $m −o image_2.iso \
      −J −graft−points /dir2=./tree2
```

Now burn the new session onto the same medium. This time without blanking:

```
$ xorrecord −v dev=/dev/sr0 speed=12 fs=8m \
      −multi −−grow_overwriteable_iso \
      −eject padsize=300k image_2.iso
```

Operating systems which mount this medium will read the superblock of the second session and show both directories /dir1 and /dir2.

Write ISO 9660 session on-the-fly:

It is possible to combine the run of **xorrisofs** and **xorrecord** in a pipeline without storing the ISO 9660 image as file on hard disk.

The piped run is more vulnerable to the problem that some systems have not enough patience with automatic tray loading and that demons may interfere with a first CD−ROM driver read attempt from a freshly loaded medium. It is advised to load the tray manually or via a separate run of xorriso with a subsequent run of dd.

Again, xorriso has the patience and dd is a dispensable victim for demons.

```
$ m=$(xorrecord dev=/dev/sr0 −msinfo)
$ dd if=/dev/sr0 count=1 >/dev/null 2>&1
$ xorrisofs −M /dev/sr0 −C $m  \
      −J −graft−points /dir2=./tree2 \
  | xorrecord −v dev=/dev/sr0 speed=12 fs=8m \
        −waiti −multi −−grow_overwriteable_iso \
        −eject padsize=300k −
```

This is also the main use case of program **xorriso** itself, where the run would need no system workarounds and simply look like:

```
$ xorriso −dev /dev/sr0 −joliet on −speed 12 −fs 8m \
     −map ./tree2 /dir2 −commit_eject all
```

Write compressed afio archive on-the-fly:

This is possible with all media except minimally blanked DVD−RW and DVD−R DL. Since the compressed output stream is of very variable speed, a larger fifo is advised. Nevertheless, this example is not suitable for very old CD drives which have no underrun protection and thus would abort the burn run on temporary data shortage.

```
$ find . | afio −oZ − | \
  xorrecord −v dev=/dev/sr0 speed=12 fs=64m \
      −multi padsize=300k −
```

afio archives do not contain references to absolute data block addresses. So they need no special precautions for multi−session. One may get the session start addresses by option −toc, and then use dd option skip= to begin reading at one of those addresses. E.g. for listing its content:

```
$ dd if=/dev/sr0 bs=2048 skip=64046 | afio −tvZ −
```

afio will know when the end of the archive is reached.

FILES

Startup files:

If not −−no_rc is given as the first argument then **xorrecord** attempts on startup to read and execute lines from the following files:

/etc/default/xorriso
/etc/opt/xorriso/rc
/etc/xorriso/xorriso.conf
$HOME/.xorrisorc

The files are read in the sequence given here, but none of them is required to exist. The lines are not interpreted as **xorrecord** options but as generic **xorriso** commands. See man xorriso.

SEE ALSO

For generic xorriso command mode

xorriso(1)

Formatting track sources for xorrecord:

xorrisofs(1), mkisofs(8), genisoimage(8), afio(1), star(1)

Other programs which burn sessions to optical media

growisofs(1), cdrecord(1), wodim(1), cdrskin(1)

BUGS

To report bugs, request help, or suggest enhancements for **xorriso**, please send electronic mail to the public list <bug−xorriso@gnu.org>. If more privacy is desired, mail to <scdbackup@gmx.net>.

Please describe what you expect **xorriso** to do, the program arguments or dialog commands by which you tried to achieve it, the messages of **xorriso**, and the undesirable outcome of your program run.

Expect to get asked more questions before solutions can be proposed.

AUTHOR

Thomas Schmitt <scdbackup@gmx.net>
for libburnia−project.org

COPYRIGHT

CREDITS

xorriso is in part based on work by Vreixo Formoso who provides libisofs together with Mario Danic who also leads the libburnia team. Thanks to Andy Polyakov who invented emulated growing, to Derek Foreman and Ben Jansens who once founded libburn.

Compliments towards Joerg Schilling whose cdrtools served me for ten years.