

# IDE-CD driver documentation

**Originally by:** scott snyder <snyder@fnald0.fnal.gov> (19 May 1996)

**Carrying on the torch is:** Erik Andersen <andersee@debian.org>

**New maintainers (19 Oct 1998):** Jens Axboe <axboe@image.dk>

## 1. Introduction

The ide-cd driver should work with all ATAPI ver 1.2 to ATAPI 2.6 compliant CDROM drives which attach to an IDE interface. Note that some CDROM vendors (including Mitsumi, Sony, Creative, Aztech, and Goldstar) have made both ATAPI-compliant drives and drives which use a proprietary interface. If your drive uses one of those proprietary interfaces, this driver will not work with it (but one of the other CDROM drivers probably will). This driver will not work with *ATAPI* drives which attach to the parallel port. In addition, there is at least one drive (CyCDROM CR520ie) which attaches to the IDE port but is not ATAPI; this driver will not work with drives like that either (but see the *aztcd* driver).

This driver provides the following features:

- Reading from data tracks, and mounting ISO 9660 filesystems.
- Playing audio tracks. Most of the CDROM player programs floating around should work; I usually use Workman.
- Multisession support.
- On drives which support it, reading digital audio data directly from audio tracks. The program *cdda2wav* can be used for this. Note, however, that only some drives actually support this.
- There is now support for CDROM changers which comply with the ATAPI 2.6 draft standard (such as the NEC CDR-251). This additional functionality includes a function call to query which slot is the currently selected slot, a function call to query which slots contain CDs, etc. A sample program which demonstrates this functionality is appended to the end of this file. The Sanyo 3-disc changer (which does not conform to the standard) is also now supported. Please note the driver refers to the first CD as slot # 0.

## 2. Installation

0. The ide-cd relies on the ide disk driver. See Documentation/ide/ide.rst for up-to-date information on the ide driver.
1. Make sure that the ide and ide-cd drivers are compiled into the kernel you're using. When configuring the kernel, in the section entitled "Floppy, IDE, and other block devices", say either *Y* (which will compile the support directly into the kernel) or *M* (to compile support as a module which can be loaded and unloaded) to the options:

```
ATA/ATAPI/MFM/RLL support
Include IDE/ATAPI CDROM support
```

Depending on what type of IDE interface you have, you may need to specify additional configuration options. See Documentation/ide/ide.rst.

2. You should also ensure that the iso9660 filesystem is either compiled into the kernel or available as a loadable module. You can see if a filesystem is known to the kernel by catting /proc/filesystems.
3. The CDROM drive should be connected to the host on an IDE interface. Each interface on a system is defined by an I/O port address and an IRQ number, the standard assignments being 0x1f0 and 14 for the primary interface and 0x170 and 15 for the secondary interface. Each interface can control up to two devices, where each device can be a hard drive, a CDROM drive, a floppy drive, or a tape drive. The two devices on an interface are called *master* and *slave*; this is usually selectable via a jumper on the drive.

Linux names these devices as follows. The master and slave devices on the primary IDE interface are called *hda* and *hdb*, respectively. The drives on the secondary interface are called *hdc* and *hdd*. (Interfaces at other locations get other letters in the third position; see Documentation/ide/ide.rst.)

If you want your CDROM drive to be found automatically by the driver, you should make sure your IDE interface uses either the primary or secondary addresses mentioned above. In addition, if the CDROM drive is the only device on the IDE interface, it should be jumpered as *master*. (If for some reason you cannot configure your system in this manner, you can probably still use the driver. You may have to pass extra configuration information to the kernel when you boot, however. See Documentation/ide/ide.rst for more information.)

4. Boot the system. If the drive is recognized, you should see a message which looks like:

```
hdb: NEC CD-ROM DRIVE:260, ATAPI CDROM drive
```

If you do not see this, see section 5 below.

5. You may want to create a symbolic link /dev/cdrom pointing to the actual device. You can do this with the command:

```
ln -s /dev/hdX /dev/cdrom
```

where X should be replaced by the letter indicating where your drive is installed.

6. You should be able to see any error messages from the driver with the *dmesg* command.

### 3. Basic usage

An ISO 9660 CDROM can be mounted by putting the disc in the drive and typing (as root):

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

where it is assumed that */dev/cdrom* is a link pointing to the actual device (as described in step 5 of the last section) and */mnt/cdrom* is an empty directory. You should now be able to see the contents of the CDROM under the */mnt/cdrom* directory. If you want to eject the CDROM, you must first dismount it with a command like:

```
umount /mnt/cdrom
```

Note that audio CDs cannot be mounted.

Some distributions set up */etc/fstab* to always try to mount a CDROM filesystem on bootup. It is not required to mount the CDROM in this manner, though, and it may be a nuisance if you change CDROMs often. You should feel free to remove the *cdrom* line from */etc/fstab* and mount CDROMs manually if that suits you better.

Multisession and photocd discs should work with no special handling. The *hpcdtoppm* package ([ftp.gwdg.de/pub/linux/hpcdtoppm/](http://ftp.gwdg.de/pub/linux/hpcdtoppm/)) may be useful for reading photocds.

To play an audio CD, you should first unmount and remove any data CDROM. Any of the CDROM player programs should then work (*workman*, *workbone*, *cdplayer*, etc.).

On a few drives, you can read digital audio directly using a program such as *cdda2wav*. The only types of drive which I've heard support this are Sony and Toshiba drives. You will get errors if you try to use this function on a drive which does not support it.

For supported changers, you can use the *cdchange* program (appended to the end of this file) to switch between changer slots. Note that the drive should be unmounted before attempting this. The program takes two arguments: the CDROM device, and the slot number to which you wish to change. If the slot number is -1, the drive is unloaded.

### 4. Common problems

This section discusses some common problems encountered when trying to use the driver, and some possible solutions. Note that if you are experiencing problems, you should probably also review *Documentation/ide/ide.rst* for current information about the underlying IDE support code. Some of these items apply only to earlier versions of the driver, but are mentioned here for completeness.

In most cases, you should probably check with *dmesg* for any errors from the driver.

- a. Drive is not detected during booting.
  - Review the configuration instructions above and in *Documentation/ide/ide.rst*, and check how your hardware is configured.
  - If your drive is the only device on an IDE interface, it should be jumpered as master, if at all possible.
  - If your IDE interface is not at the standard addresses of 0x170 or 0x1f0, you'll need to explicitly inform the driver using a *lilo* option. See *Documentation/ide/ide.rst*. (This feature was added around kernel version 1.3.30.)
  - If the autoprobing is not finding your drive, you can tell the driver to assume that one exists by using a *lilo* option of the form *hdX=cdrom*, where X is the drive letter corresponding to where your drive is installed. Note that if you do this and you see a boot message like:

```
hdX: ATAPI cdrom (?)
```

this does not mean that the driver has successfully detected the drive; rather, it means that the driver has not detected a drive, but is assuming there's one there anyway because you told it so. If you actually try to do I/O to a drive defined at a nonexistent or nonresponding I/O address, you'll probably get errors with a status value of 0xff.

- Some IDE adapters require a nonstandard initialization sequence before they'll function properly. (If this is the case, there will often be a separate MS-DOS driver just for the controller.) IDE interfaces on sound cards often fall into this category.

Support for some interfaces needing extra initialization is provided in later 1.3.x kernels. You may need to turn on additional kernel configuration options to get them to work; see *Documentation/ide/ide.rst*.

Even if support is not available for your interface, you may be able to get it to work with the following procedure. First boot MS-DOS and load the appropriate drivers. Then warm-boot linux (i.e., without powering off). If this works, it can be automated by running *loadlin* from the MS-DOS *autoexec*.

- b. Timeout/IRQ errors.

- If you always get timeout errors, interrupts from the drive are probably not making it to the host.
- IRQ problems may also be indicated by the message *IRQ probe failed (<n>)* while booting. If <n> is zero, that means that the system did not see an interrupt from the drive when it was expecting one (on any feasible IRQ). If <n> is negative, that means the system saw interrupts on multiple IRQ lines, when it was expecting to receive just one from the CDROM drive.
- Double-check your hardware configuration to make sure that the IRQ number of your IDE interface matches what the driver expects. (The usual assignments are 14 for the primary (0x1f0) interface and 15 for the secondary (0x170) interface.) Also be sure that you don't have some other hardware which might be conflicting with the IRQ you're using. Also check the BIOS setup for your system; some have the ability to disable individual IRQ levels, and I've had one report of a system which was shipped with IRQ 15 disabled by default.
- Note that many MS-DOS CDROM drivers will still function even if there are hardware problems with the interrupt setup; they apparently don't use interrupts.
- If you own a Pioneer DR-A24X, you \_will\_ get nasty error messages on boot such as "irq timeout: status=0x50 { DriveReady SeekComplete }". The Pioneer DR-A24X CDROM drives are fairly popular these days. Unfortunately, these drives seem to become very confused when we perform the standard Linux ATA disk drive probe. If you own one of these drives, you can bypass the ATA probing which confuses these CDROM drives, by adding *append="hdX=noprobe hdX=cdrom"* to your lilo.conf file and running lilo (again where X is the drive letter corresponding to where your drive is installed.)

c. System hangups.

- If the system locks up when you try to access the CDROM, the most likely cause is that you have a buggy IDE adapter which doesn't properly handle simultaneous transactions on multiple interfaces. The most notorious of these is the CMD640B chip. This problem can be worked around by specifying the *serialize* option when booting. Recent kernels should be able to detect the need for this automatically in most cases, but the detection is not foolproof. See Documentation/ide/ide.rst for more information about the *serialize* option and the CMD640B.
- Note that many MS-DOS CDROM drivers will work with such buggy hardware, apparently because they never attempt to overlap CDROM operations with other disk activity.

d. Can't mount a CDROM.

- If you get errors from mount, it may help to check *dmesg* to see if there are any more specific errors from the driver or from the filesystem.
- Make sure there's a CDROM loaded in the drive, and that it's an ISO 9660 disc. You can't mount an audio CD.
- With the CDROM in the drive and unmounted, try something like:

```
cat /dev/cdrom | od | more
```

If you see a dump, then the drive and driver are probably working OK, and the problem is at the filesystem level (i.e., the CDROM is not ISO 9660 or has errors in the filesystem structure).

- If you see *not a block device* errors, check that the definitions of the device special files are correct. They should be as follows:

```
brw-rw---- 1 root    disk      3,    0 Nov 11 18:48 /dev/hda
brw-rw---- 1 root    disk      3,   64 Nov 11 18:48 /dev/hdb
brw-rw---- 1 root    disk     22,    0 Nov 11 18:48 /dev/hdc
brw-rw---- 1 root    disk     22,   64 Nov 11 18:48 /dev/hdd
```

Some early Slackware releases had these defined incorrectly. If these are wrong, you can remake them by running the script *scripts/MAKEDEV.ide*. (You may have to make it executable with *chmod* first.)

If you have a */dev/cdrom* symbolic link, check that it is pointing to the correct device file.

If you hear people talking of the devices *hd1a* and *hd1b*, these were old names for what are now called *hdc* and *hdd*. Those names should be considered obsolete.

- If mount is complaining that the iso9660 filesystem is not available, but you know it is (check */proc/filesystems*), you probably need a newer version of mount. Early versions would not always give meaningful error messages.

e. Directory listings are unpredictably truncated, and *dmesg* shows *buffer botch* error messages from the driver.

- There was a bug in the version of the driver in 1.2.x kernels which could cause this. It was fixed in 1.3.0. If you can't upgrade, you can probably work around the problem by specifying a blocksize of 2048 when mounting. (Note that you won't be able to directly execute binaries off the CDROM in that case.)

If you see this in kernels later than 1.3.0, please report it as a bug.

f. Data corruption.

- Random data corruption was occasionally observed with the Hitachi CDR-7730 CDROM. If you experience data

corruption, using "hdx=slow" as a command line parameter may work around the problem, at the expense of low system performance.

## 5. cdchange.c

```
/*
 * cdchange.c [-v] <device> [<slot>]
 *
 * This loads a CDROM from a specified slot in a changer, and displays
 * information about the changer status. The drive should be unmounted before
 * using this program.
 *
 * Changer information is displayed if either the -v flag is specified
 * or no slot was specified.
 *
 * Based on code originally from Gerhard Zuber <zuber@berlin.snafu.de>.
 * Changer status information, and rewrite for the new Uniform CDROM driver
 * interface by Erik Andersen <andersee@debian.org>.
 */

#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/cdrom.h>

int
main (int argc, char **argv)
{
    char *program;
    char *device;
    int fd;           /* file descriptor for CD-ROM device */
    int status;       /* return status for system calls */
    int verbose = 0;
    int slot=-1, x_slot;
    int total_slots_available;

    program = argv[0];

    ++argv;
    --argc;

    if (argc < 1 || argc > 3) {
        fprintf (stderr, "usage: %s [-v] <device> [<slot>]\n",
                 program);
        fprintf (stderr, "          Slots are numbered 1 -- n.\n");
        exit (1);
    }

    if (strcmp (argv[0], "-v") == 0) {
        verbose = 1;
        ++argv;
        --argc;
    }

    device = argv[0];

    if (argc == 2)
        slot = atoi (argv[1]) - 1;

    /* open device */
    fd = open(device, O_RDONLY | O_NONBLOCK);
    if (fd < 0) {
        fprintf (stderr, "%s: open failed for `%s`: %s\n",
                 program, device, strerror (errno));
        exit (1);
    }

    /* Check CD player status */
    total_slots_available = ioctl (fd, CDROM_CHANGER_NSLOTS);
    if (total_slots_available <= 1) {
        fprintf (stderr, "%s: Device `%s` is not an ATAPI "
                 "compliant CD changer.\n", program, device);
        exit (1);
    }
}
```

```

if (slot >= 0) {
    if (slot >= total_slots_available) {
        fprintf (stderr, "Bad slot number.  "
                  "Should be 1 -- %d.\n",
                  total_slots_available);
        exit (1);
    }

    /* load */
    slot=iocctl (fd, CDROM_SELECT_DISC, slot);
    if (slot<0) {
        fflush(stdout);
        perror ("CDROM_SELECT_DISC ");
        exit(1);
    }
}

if (slot < 0 || verbose) {

    status=iocctl (fd, CDROM_SELECT_DISC, CDSL_CURRENT);
    if (status<0) {
        fflush(stdout);
        perror (" CDROM_SELECT_DISC");
        exit(1);
    }
    slot=status;

    printf ("Current slot: %d\n", slot+1);
    printf ("Total slots available: %d\n",
            total_slots_available);

    printf ("Drive status: ");
    status = iocctl (fd, CDROM_DRIVE_STATUS, CDSL_CURRENT);
    if (status<0) {
        perror(" CDROM_DRIVE_STATUS");
    } else switch(status) {
    case CDS_DISC_OK:
        printf ("Ready.\n");
        break;
    case CDS_TRAY_OPEN:
        printf ("Tray Open.\n");
        break;
    case CDS_DRIVE_NOT_READY:
        printf ("Drive Not Ready.\n");
        break;
    default:
        printf ("This Should not happen!\n");
        break;
    }

    for (x_slot=0; x_slot<total_slots_available; x_slot++) {
        printf ("Slot %2d: ", x_slot+1);
        status = iocctl (fd, CDROM_DRIVE_STATUS, x_slot);
        if (status<0) {
            perror(" CDROM_DRIVE_STATUS");
        } else switch(status) {
        case CDS_DISC_OK:
            printf ("Disc present.");
            break;
        case CDS_NO_DISC:
            printf ("Empty slot.");
            break;
        case CDS_TRAY_OPEN:
            printf ("CD-ROM tray open.\n");
            break;
        case CDS_DRIVE_NOT_READY:
            printf ("CD-ROM drive not ready.\n");
            break;
        case CDS_NO_INFO:
            printf ("No Information available.");
            break;
        default:
            printf ("This Should not happen!\n");
            break;
        }
        if (slot == x_slot) {
            status = iocctl (fd, CDROM_DISC_STATUS);
            if (status<0) {
                perror(" CDROM_DISC_STATUS");
            }
            switch (status) {

```

```

        case CDS_AUDIO:
            printf ("\tAudio disc.\t");
            break;
        case CDS_DATA_1:
        case CDS_DATA_2:
            printf ("\tData disc type %d.\t", status-CDS_DATA_1+1);
            break;
        case CDS_XA_2_1:
        case CDS_XA_2_2:
            printf ("\tXA data disc type %d.\t", status-CDS_XA_2_1+1);
            break;
        default:
            printf ("\tUnknown disc type 0x%x!\t", status);
            break;
    }
}
status = ioctl (fd, CDROM_MEDIA_CHANGED, x_slot);
if (status<0) {
    perror(" CDROM_MEDIA_CHANGED");
}
switch (status) {
case 1:
    printf ("Changed.\n");
    break;
default:
    printf ("\n");
    break;
}
}

/* close device */
status = close (fd);
if (status != 0) {
    fprintf (stderr, "%s: close failed for `%s`: %s\n",
            program, device, strerror (errno));
    exit (1);
}

exit (0);
}

```