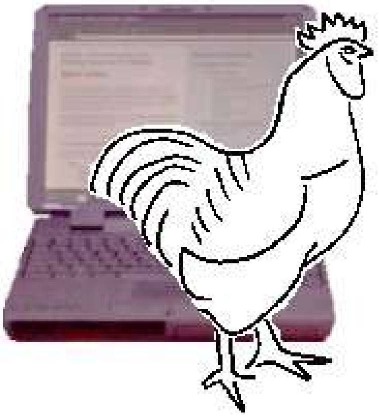
The Community–Linux Training Centre Project:

USB Mass Storage Devices

CLTC Building Blocks Briefing 1

By the Free Range Community–Linux Training Centre Project.

CLTC_BB-01, version 1.0, January 2004 http://www.fraw.org.uk/cltc/



Today many of the devices used with computers – such as digital cameras – store information. Most use the USB port to transfer this information to your computer. This briefing looks at how to read data from USB storage devices.

USB – the *Universal Serial Bus* – was developed to allow peripherals to be plugged easily into computer systems. It was faster than old *9-pin D-type* RS-232 serial ports that computers used and it allowed many devices to share the same port via a hub. The first version of the USB was soon challenged by the much faster alternative *Firewire* (also called *IEEE-1394*) system. The recent revision to the USB standard, USB-2, has increased the speed at which USB-2 enabled devices can operate. But Firewire has become the standard for moving large quantities of data, e.g. digital video (DV) cameras.

USB devices fall into two categories. Some process data instantaneously, like scanners or modems. Other store data, transferring it to the computer on demand. It's this second category of devices that this briefing addresses.

Problem: Is it a Mass Storage Device?

Connecting a USB device to a Linux system can be very simple if it uses the mass storage device standard. Many of the early digital cameras used their own proprietary format for exchanging data. This makes it difficult to connect the device to a Linux system unless someone has written a filter program to process the information – for example, the `gphoto` program that allows different digital cameras to interface with the Linux system.

Mass storage devices (MSDs) allow data to be accessed via USB as if it were another hard or floppy disk connected to the computer system. Therefore you can access the content of the device directly as a collection of files rather than via a special program. There are two ways to tell if the device is a MSD device that works with Linux:

1. *Plug it into the USB port* – if it works you should be able to use it straight away (see ' mounting file systems' below).
2. *Try to manually mount the device* – checking with the Linux system as to whether it recognises the device or not.

Some devices may still present problems. In this case, if the device uses removable storage media like *smart media*, *compact flash* or a *micro-drive*, your simplest option is to buy a storage card reader. This allows you to take the card out of the storage device and access it directly from the Linux system.

Mounting the File System

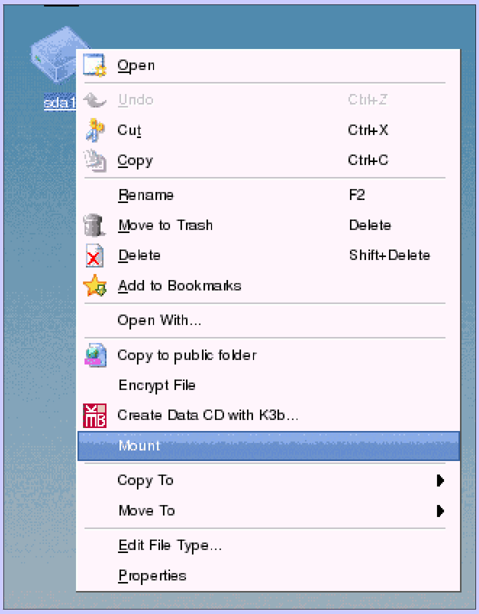
To store information computers use *file systems* – a set of standard methods to hold files and make them available to other programs. To be compatible with computers, most mass storage devices encode information into a common file system format – by default a Microsoft *Windows* compatible format. Linux can usually access all these various file systems and exchange information with them.

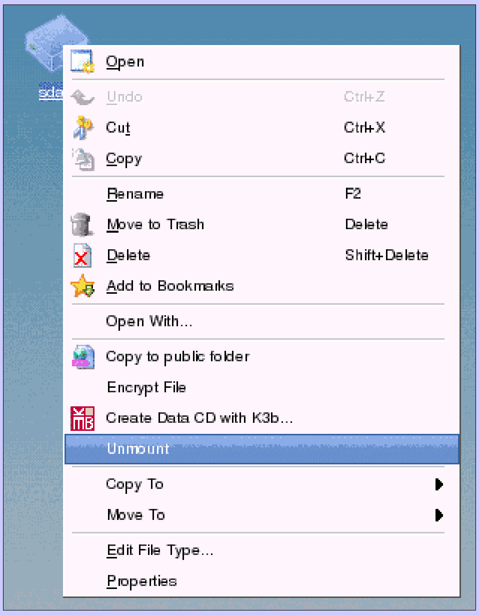
Most of the recent Linux distributions, that support the *hot-plugging* of USB, automatically detect when you plug a device into one of the USB sockets. The devices attached to the USB hub on your computer are usually dealt with as *SCSI* devices, so the system gives them a name such as `/dev/sda`. When the Linux system has this sorted out this identity it can address the memory inside the device and exchange information.

Mounting/Unmounting the Device

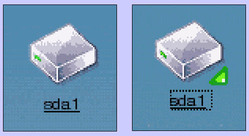
The following shows mounting and unmounting of the storage device file system using KDE and Gnome.

Using KDE: Right click on the desktop icon to access the mount and unmount actions, and then click on the mounted icon to get a file window.

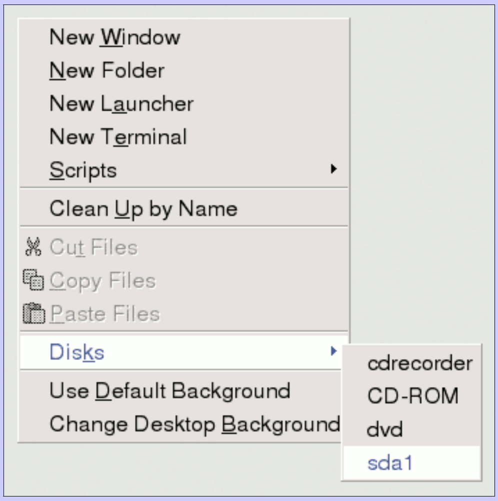


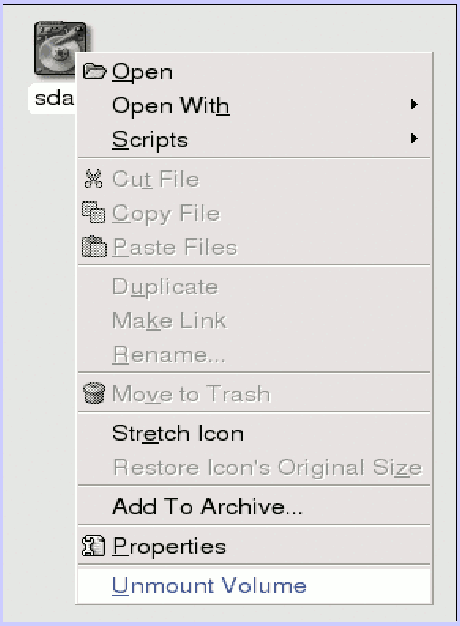


Note also the difference between the unmounted (left) and mounted (right) icons.



Using Gnome: Right click on the desktop to access the disks menu, then click the drive you wish to mount. Click on the desktop icon (below) when it appears to get a file window. To unmount, right click the desktop icon and click *unmount volume*.





To *mount* the device on the file system the kernel connects a location on your normal file system to the device ID of the storage device. Then you can access the files within the device as if they were another part of you computer' s hard disk.

After inserting the USB plug it can take between 10 and 30 seconds for the device to be detected and allocated a *mount point*. What happens then depends on which desktop you are running and how old it is (see box, left):

- In *KDE*, a new icon will appear on the desktop representing the device plugged into the USB port – with a name such as `sda1`, `sdb1`, etc.
- In *Gnome*, you right click on the desktop, and then click on *Disks*, to see what storage devices are connected. If recognised, there will be a new device listed.

To mount the device you either click in the desktop icon (in *KDE*) or go to *Disks* and click on the device name (in *Gnome*). This causes the desktop icon to change from an unmounted to a mounted version (in KDE), or a new icon will appear on the desktop (Gnome). Click on the icon to bring up a file window – this allows you to access the contents of the storage device directly. From then on you use this window just like other directory windows – you can drag and drop files from or to the USB storage device.

You may find problems with some older versions of KDE and Gnome. This is because the desktop isn' t automatically presenting you with icons for hot-plugged devices. In this case you' llneed to edit the *file systems table file* (usually called `fstab`) on your system to tell the desktop to provide access to `/dev/sda1`. Instructions on how to do this are included in the next section.

USB mass storage devices

MP3 music file player (top left), 128MB flash data card (top right) and a smart media/compact flash/micro-drive card reader (bottom).

Mounting from the Command Line

You can also mount devices from the command line if you have problems mounting devices from the desktop. Connecting non-standard devices is something only that the *root user* on the Linux system can do. For this reason you' llhave to use the `su` command to temporarily turn yourself into the root user. The alternative is to tell the Linux system to include the `/dev/sda` devices as part of the resources.

To mount a device from the command line first open a terminal box (assuming you' reworking from a graphical desktop). To change yourself in to root you first type `su` and press return. Then you are asked for the root password. Enter the password, press return, and you should notice that the command prompt now ends in a '#' – meaning that you are now root.

First we need to create a mount point for the device, which means creating a directory to hold the data inside the device. For safety, is probably advisable to create the directory within the /home directory. So enter the following command (if you wish, changing ' usb'to be the name you' dlike the directory to have) and press return –

```
  mkdir /home/usb
```

Assuming that the USB storage device has been properly recognised by the Linux kernel, and has been given a `/dev/sda?` device ID, to mount the device type –

```
  mount -t auto /dev/sda1 /home/usb
```

This command links the device `/dev/sda1` to the file system at the directory `/home/usb`. The `-t auto` part tells the mount program to examine the file system and guess the file system type. If you get a command prompt after pressing return then the command was successful. If not, then you' ve got a problem (see ' troubleshooting' .)

Once mounted you can use the storage device as another part of the file system – using the command line to move data to and fro. When you have finished using the device you must unmount it from the file system using the command –

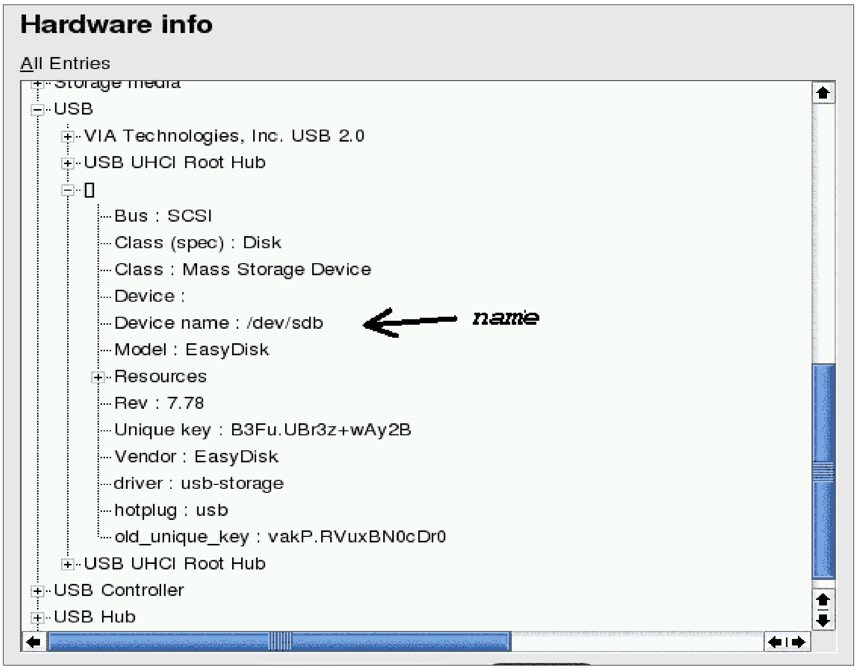
```
  umount /dev/sda1
```

To stop being root, type `exit` and press return.

If you want to make life easier you' ll need to change the file system table – `fstab`. To do this you need to log into your system as root. Open a text editor, and then open the file `/etc/fstab`. The box at the top of the next page shows what to add to the file – you should be able to see from the format of the `fstab` file how this fits in. Save the file with this new line inserted at the end, and when you restart your system you should be able to access your USB devices without having to login as root. You should also

New line for `/etc/fstab` file to access USB devices

```
/dev/sda1   /home/usb   auto   sync,noauto,user,exec  0 0
```



```
Hardware info

All Entries
+-Storage media
-USB
   +-VIA Technologies, Inc. USB 2.0
   +-USB UHCI Root Hub
   -▯
      Bus : SCSI
      Class (spec) : Disk
      Class : Mass Storage Device
      Device :
      Device name : /dev/sdb
      Model : EasyDisk
      +-Resources
      Rev : 7.78
      Unique key : B3Fu.UBr3z+wAy2B
      Vendor : EasyDisk
      driver : usb-storage
      hotplug : usb
      old_unique_key : vakP.RVuxBN0cDr0
   +-USB UHCI Root Hub
+-USB Controller
+-USB Hub
```

The SuSE Linux *YaST* hardware information window

find that your desktop will create an icon for the /dev/sda1 automatically, even when it is not plugged in.

Troubleshooting

If your system doesn' t auto-detect the USB device then you' ll have to go hunting for it. If you are using a Linux distribution such as *SuSE* or *Mandrake* you can use the hardware configuration utility to tell you whether your device has been detected (see below), and if so what name it has been given. If not then you' ll have to use a console program like `usbview` to list what devices are currently connected to the USB ports.

If your device is not listed it has not been detected, and you can' t access it. If the device is listed use the ID given in the device name line, add a ' 1' (e.g., `/dev/sdb1`) at the end, and you should be able to mount it onto the file system using this ID. However, if the device uses a non-standard file system, you may still not be able to mount and read the device.

Using USB Storage Devices

The picture on the previous page shows various USB storage devices. Although they differ in function, in practical terms they all work the same way. You mount the file system, and then read, copy and delete files just as if you were using an ordinary floppy disk.

Whilst the MP3 player or USB flash card pictured only appear as one device, the card reader at the bottom of the picture is more complex. It has three slots for different types of storage media. For this reason it appears as three devices, and you must identify which slot has which device ID – `/dev/sda1`, `/dev/sdb1` or `/dev/sdc1` – using one of the USB hardware information programs described in the 'troubleshooting' section above. You can either create mount points for each of the three devices, or just the one that you use the most. Despite this extra work, card readers are very useful gadgets. This is because many of the devices that you can' t access via USB have removable storage media. The reader enables you to access the information on the removable storage media instead of connecting to the device directly.

The Community–Linux Training Centre Project has been developed by the Free Range Network to promote the use of Gnu/Linux as a low-cost training and education resource for community and grass roots campaigning organisations. The purpose of the *building block briefings* is to cover a very small subject in detail. This allows novice Linux users, by combining the content of different building block briefings, to undertake larger projects using their system.