# Partition-Rescue

**Revision History**

| | | |
|---|---|---|
| Revision 1 | 2008-11-24 09:27:50 | Revised by: jdd |

mainly title change in the wiki

# Table of Contents

**1. Revision History....1**

**2. Beginning....2**
2.1. What's in....2
2.2. What to do right now?....2
2.3. Legal stuff....2
2.4. What do I need to know right now?....3

**3. Technical info....4**
3.1. Disks....4
3.2. Partitions....4
3.3. Why is there a problem?....5

**4. Solving the problem....6**
4.1. The simpler case....6
4.2. A not-so-simple case....7
4.2.1. By hand....7
4.2.2. Linux's own info and other hacks....7
4.3. The rich man's case....10
4.4. PMagic....10

**5. References....11**
5.1. Authors....11
5.2. Most recent version....11

# 1. Revision History

Revision V4.1 - 2008-11-24 - Revised by jdd

Title fix (the wiki have to get the exact name of the HOWTO)

Revision v4 - 2008-08-29 - Revised by: jdd

Major update - First complete revised wiki version

Revision v3.7 - 2008-05-30 - Revised by: rm

Minor update - lots of little punctuation, spelling, usage, and grammar nits :-)

Revision v3.6 - 2008-05-25 - Revised by: jdd

Major update - LDP wiki - addition about logical partitions

Revision v3.5 - 2003-10-31 - Revised by: jdd

Major update - new licence - some fixes in addresses - vi use :-)

Revision v3.4 - 2002-08-22 - Revised by: jdd

Minor update related only with docbook

Revision v3.3 - 2001-11-17 - Revised by: jdd

Minor update - docbook & revision history - emacs use.

Revision v3.2 - 2001-09-25 - Revised by: jdd

Major update.

Whow! My disk is empty! My Linux is gone! If you have or fear to have one day or another such a problem, read this...

---

# 2. Beginning

## 2.1. What's in

This HOWTO addresses only the "lost partition table" problem. This can be when:

- you have no more access to your computer, with the "no operating system" message;
- you have installed a new system (i.e., MS Windows) and you see no more Linux, and MS Windows takes up all the capacity of the disk;
- you have just partitioned the wrong drive with fdisk (for example, in the process of changing your hard drive).

Here, you will learn that, if you know the right thing and do it, Linux comes usually safe from such things. MS Windows can, but it's luckier.

We will first see what you can do *before* the problem to ease future recovery, and what you must do *after* to recover. There is little to do to prevent from erasing a disk; usually this is done by automatic MS Windows or Linux-install ill-behaved programs or users' mistakes - nothing can be done to prevent this, except care, but you are already careful, aren't you?

It can also be done by the use of MS-DOS/Windows fdisk. Avoid it as most as you can, but you probably can't.

I have done this many times, on my computer and on other guys' computers, and restored Linux most of the time and MS Windows sometimes. I wish you luck!

---

## 2.2. What to do right now?

If you don't have any problem yet, if you read this by curiosity or are just seeking information, and you are on a running Linux system, do immediately the following :

- open a root terminal or xterm;
- key in /sbin/fdisk -l (that last character being l for Lima). Then do fdisk -u -l;

You will be gratified to see a list of all current partitions, on all disks present on your computer. The second one gives the listing in units of sectors, in place of cylinders, and this is sometimes necessary.

- Write this back on paper (or do "/sbin/fdisk -l | lpr" and "/sbin/fdisk -u -l | lpr" to print it) and save it in a safe place for future use. If you are not the system administrator, you should not be concerned by the problem, and can stop reading this.

---

## 2.3. Legal stuff

This HOWTO is Copyright (c) 2000-2008 by Jean-Daniel Dodin. As of November 2003, the licence is LGPL.

I am not responsible of any damage on any computer as a result of anyone reading this HOWTO. If you do any damage, *it is YOUR fault, NOT MINE*! Be careful when partitioning disks, and don't make any mistakes, because it can be fatal! Backup all your important data and check that everything you do is correct! What is

described here worked on my computer, but it may or may not work on your computer. Although it should work for everyone, I can't guarantee anything. This is the last warning you get: BACKUP IMPORTANT DATA! Or, to put it concisely: Use at your own risk!

## 2.4. What do I need to know right now?

You *need* to know that, in case of any major problem with your hard disk, you *need* to stop using it in *write* mode, at least until the time for you to understand what's happening. Information there is very volatile...

If ever, one morning, awaking, your computer says "can't load, no system installed", you *must not begin reinstalling all the stuff.*

If you have MS Windows installed, I can't promise you can recover your data, but it's likely you will recover all your Linux stuff, provided it's not located too low (near the beginning of the disk) in the disk structure. This is because some MS Windows viruses erase the very first disk cylinder, whatever is on. However I didn't ever experiment with such viruses, and can't say for sure. Try recovering, anyway.

You must also know that I give you all this information only for this - information purposes. Neither I nor any other people but you can be held responsible for any problem your data can have using this info. There are too many different systems on the world for anybody being able to promise anything. I can only wish you luck and hope that you, like me, will be happy recovering data.

# 3. Technical info

## 3.1. Disks

A hard disk is made of sectors numbered from 0 to the max.

dmesg gives, for example:

```
hdb: ST34321A, 4103MB w/128kB Cache, CHS=523/255/63
```

CHS means Cylinders, Heads, Sectors.

523*255*63=8401995 sectors of 512 bytes, thus the 4103 MB. This is only a logical map; it's not necessarily what is written on the disk cover (except for the total size).

The true size of the sectors is of no interest for us, given that we don't want to modify anything, but merely wish to restore a previous state. For us, the default size given by fdisk is all right.

The size seen by the system is directly dependent on the work of the BIOS (Basic Input/Output System - the PC's ROM). The mode of the hard disk indicated in the BIOS is essential. On a new disk, it's better to use BIOS automatic hard disk recognition and say "yes". Anyway, any modification at this level may destroy all the data of the disk, so don't play with this without essential reason.

This is probably what your disk already uses, so don't be afraid.

---

## 3.2. Partitions

Disk are now huge -- 500 GB drives are not rare -- so it's not really handy to have all this stuff packed in only one part. Only MS Windows does so, and, if you use Linux, maybe it's because you are aware of how inefficient the other is.

So a hard disk is usually cut in some pieces called "partitions" (see the Partition HOWTO for details, also read *README.fdisk* on the web or on your disk - location vary).

Let's get a look at (part of) my own print of fdisk -l :

```
Disk /dev/hdb: 255 heads, 63 sectors, 523 cylinders
Units = cylinders of 16065 * 512 bytes
Device Boot Start End Blocks Id System
/dev/hdb1 1 153 1228941 83 Linux
/dev/hdb2 154 166 104422+ 82 Linux swap
/dev/hdb3 * 167 291 1004062+ 83 Linux
/dev/hdb4 295 523 1839442+ 5 Extended
/dev/hdb5 295 422 1028128+ 83 Linux
/dev/hdb6 423 523 811251 6 FAT16
```

This is my second hard disk, tied to guesses and tries. (The first is too simple to be interesting.)

/dev/hdb is my second ide disk (slave on the primary interface),

/dev/hdb1 is the first primary partition, running from the first (1) block to the block 153.

There can be four such primary partitions. If one wants more than 4, one of them must be repurposed as an "extended" partition (not necessarily the fourth), and all other partitions are "logical" and are located *inside* the extended one. Notice that partition number 5 and partition number 4 have the same beginning. Number five is logical, number 4 extended. Logical partitions' numbering always begins at 5, even if there are only 2 primary partitions.

Here's the fdisk -u -l listing of an other disk:

```
Disque /dev/hda : 240 tÃªtes, 63 secteurs, 2584
cylindres UnitÃ©s = secteurs sur 1 * 512 octets
PÃ©riphÃ©rique Amorce DÃ©but Fin Blocs Id SystÃ¨me
/dev/hda1 * 63 10357199 5178568+ c Win95 FAT32 (LBA)
/dev/hda2 15452640 39070079 11808720 83 Linux
/dev/hda3 10357200 15150239 2396520 f Win95 Etdue (LBA)
/dev/hda4 15150240 15452639 151200 84 Lecteur C: cachÃ© OS/2
/dev/hda5 10357263 10463039 52888+ 83 Linux
/dev/hda6 10463103 10780559 158728+ 82 Echange Linux
/dev/hda7 10780623 15150239 2184808+ 6 FAT16
Les entrÃ©es de la table de partitions ne suivent pas l'ordre du disque.
```

Don't worry about the French part, I'm French ... look at your own disk listing. Of course, numbers are bigger.

## 3.3. Why is there a problem?

The problem is that all installed operating systems must share the disks, and, since at start, the BIOS only scans the first one, there must be a so called "partition table" at the very beginning of this disk. This partition table is located in the Master Boot Record (MBR), side by side with the boot loader.

Any misuse of the MBR by any of the OS's leads to problems. When trying to install any system, a "yes" answer at a question like "automatic partitioning?" is likely to give problems... This is specially true with MS Windows, and especially with custom MS Windows installations made by special makes' PCs (when no true "Windows" CD is included, as with many laptops). But it's also true with some "smart" (not so smart!) Linux installation programs included with some distributions (hopefully this is not more the case in 2008).

# 4. Solving the problem

Please, beware! Following the explanations given here will lead you to revert back to a previous system, losing all your recent changes, if any! You must choose...

## 4.1. The simpler case

All is simple if you have at hand:

- A disk (floppy, usb key or CD) able to start Linux by itself with fdisk available - most rescue disks of any distribution can do that;
- A paper with the fdisk -l and fdisk -u -l content written down.

It's enough to:

1. Start Linux;
2. Start fdisk /dev/hda (or whatever is the disk to rescue);
3. Use fdisk to delete (d option) all the existing partitions on the damaged disk;
4. Use fdisk to create all the primary (1 - 4) partitions mentioned on the paper;
5. Give them the appropriate tag (t option) : 82 is for Linux swap, 83 for Linux main (L gives you the list), 5 is extended and must be done before creating logical partitions, c is MS Windows FAT32, and f is MS Windows extended when 6 is MS Windows FAT16.
6. Create any logical partition.

On my SUSE installation and any time I had to do this for other people, this has produced good results.

However, I said that some fdisks may cut partitions on a sector basis, not cylinder. So the fdisk -u -l version of the paper.

For using the fdisk -u -l listing one must start fdisk -u :-). In my opinion, using sector limit is a very bad idea, but it may have a real use I'm not aware of. The problem is that, with cylinder limit, it's easy to guess even if you don't have paper. With sector one, there are many more guesses...

fdisk is a small and very smart programs. There are many other makes of fdisk, but I always prefer the bare bones one. (I speak of Linux ones, of course, not the others....)

Be aware that fdisk doesn't write anything to disk before you hit w and return. If you fear a mistake, hit q (quit) or Ctrl C (^C) to quit safe without saving changes.

When your new partition table is written, start your Linux. It's possible you might not be able to do that as usual: lilo/grub may have been damaged also, and you thus may need a boot floppy or CD. Choose the option "booting the installed partition".

If you are accustomed to booting with lilo, as soon as you are logged in as root, key in "lilo" and hit return to reinstall your favourite boot loader. Right now, I'm not sure that the same thing is as easy with GRUB, but it should not be very difficult, either.

Your Linux should be all here; test it. Try, also, to start MS Windows if applicable. If you can't, there is a (very small) chance you can read your data from Linux, maybe with a raw sector-by-sector read. If you can

identify the disk sectors your data is on, using dd you can copy it to a file. This is wise for text only. This recovery is NOT in the scope of this mini-HOWTO.

## 4.2. A not-so-simple case

### 4.2.1. By hand

This is when the previous case can't be used, for lack of fdisk paper, or if it won't work for use of an out-of-date one.

**Warning**

You can only try *primary* partitions with no fear. *logical* partition uses ordinary sectors of the disk to store their own housekeeping data, so, each time you write some logical partition with fdisk, you write some sectors, erasing the data content, if any. There is still a chance you don't have any data there or the data is unimportant, but, the less often you do such tries, the better.

First, be aware that as soon as you don't write to the disk (except with fdisk), you don't erase your data, so that you can use a block-by-block try. That is you need to know the beginning of the partition to start it. If, say a 153 doesn't fit, try a 154, and so on.

This can be tiresome, but, if you remember approximately the size of the Linux partition, there is a chance to win.

### 4.2.2. Linux's own info and other hacks

#### 4.2.2.1. Kernel

If you just destroyed your own partition table, but have not rebooted Linux: Don't reboot! You can still retrieve the partition information stored in the Kernel:

cat /proc/partitions gives

| major | minor | #blocks | name |
|---|---|---|---|
| 3 | 0 | 19535040 | hda |
| 3 | 1 | 2096451 | hda1 |
| 3 | 2 | 4980150 | hda2 |
| 3 | 3 | 1 | hda3 <--- this marks an extended partition |
| 3 | 5 | 4980118 | hda5 |
| 3 | 6 | 4972086 | hda6 |

#### 4.2.2.2. hdparm

hdparm -g /dev/hda1/dev/hda1 :

geometry = 2432/255/63, sectors = 4192902, start = 63

You'll need to do a few unit conversions. "blocks" are usually 1K in length. "Sectors" are disk sectors, often 512 bytes. But usually the disk partitioning tools work in units of cylinders. (Here, 255*63=16065 sectors.) Using this information, you can build a new partition table.

---

#### 4.2.2.3. I know the start of the partition, but not the end.

If you know the start of a Linux partition, but not the end, you can still mount it, and learn about the structure. Set the partition table start correctly, and set the end to something very large. See if you guessed correctly with:

```
e2fsck -n /dev/hd??
```

You can even mount the partition and check the size:

```
mount -r /dev/hd?? /mnt
df -T
```

This won't directly tell you where the next partition starts, because of rounding issues. But it can help you get close. Be sure to use the "-n" and "-r" flags, to treat the system as read-only!!!

---

#### 4.2.2.4. Other places partition information is stored

Some distributions record partition information in a file. Of course, you probably won't be able to get to this file when you need it. But, just in case:

```
SuSE: /var/lib/YaST/install.inf
```

(if you are aware of others, please e-mail the maintainer of this document)

---

#### 4.2.2.5. gpart

But there is a better way if you can still access the Net or have "gpart" on hand. gpart is available in most distribution, at http://freshmeat.net/ or directly from http://www.stud.uni-hannover.de/user/76201/gpart.

Please note that gpart is **not** gparted - the GNOME partition editor.

"gpart - guess PC-type hard disk partitions" as the first line of the it's man page states (man gpart). And goes on to say:

"gpart tries to guess which partitions are on a hard disk. If the primary partition table has been lost, overwritten, or destroyed, the partitions still exist on the disk, but the operating system cannot access them."

This is exactly what we need. gpart is a very useful tool.

The problem is the following: the first block of any partition is marked. But it's never "unmarked" if not overwritten. So, many "first partition block" exist on an old disk, and gpart tries to do its best guessing what is the good one. In fact, it's not too difficult to try; nothing is written on the disk by gpart.

Here is the result of gpart on the previously seen disk, hdb:

```
root@charles:/home/jdd > gpart /dev/hdb
Begin scan...
Possible partition(Linux ext2), size(1200Mb), offset(0Mb)
Possible partition(Windows NTFS), size(1200Mb), offset(1200Mb)
```

```
Possible partition(Linux ext2), size(1004Mb), offset(2402Mb)
Possible partition(Windows NTFS), size(1600Mb), offset(4102Mb)
End scan.
Checking partitions...
* Warning: partition(OS/2 HPFS, NTFS, QNX or Advanced UNIX) ends beyond disk end .
Partition(Linux ext2 filesystem): primary
Partition(OS/2 HPFS, NTFS, QNX or Advanced UNIX): primary
Partition(Linux ext2 filesystem): primary
Partition(OS/2 HPFS, NTFS, QNX or Advanced UNIX): invalid primary
Ok.
Guessed primary partition table:
Primary partition(1)
type: 131(0x83)(Linux ext2 filesystem)
size: 1200mb #s(2457880) s(63-2457942)
chs: (0/1/1)-(152/254/61)d (0/1/1)-(152/254/61)r
Primary partition(2)
type: 007(0x07)(OS/2 HPFS, NTFS, QNX or Advanced UNIX)
size: 1200mb #s(2457880) s(2457944-4915823)
chs: (152/254/63)-(305/253/60)d (152/254/63)-(305/253/60)r
Primary partition(3)
type: 131(0x83)(Linux ext2 filesystem)
size: 1004mb #s(2056256) s(4919781-6976036)
chs: (306/61/49)-(434/60/47)d (306/61/49)-(434/60/47)r
Primary partition(4)
type: 000(0x00)(unused) size: 0mb #s(0) s(0-0) chs: (0/0/0)-(0/0/0)d (0/0/0)-(0/0/0)r
```

As you see, primary partition can be recovered, but, for extended ones, it's still to be done.

DOS partitions are labelled "Windows NTFS" because they were created while trying to install MS Windows 2000 (a very awful experience in year 2000!). The "invalid" one is, in fact the extended partition.

With this, one can use fdisk and try re-creating the partition table. (Remember, this is risk-free given the original one is already lost.)

gpart is updated on a weekly basis and so new versions may be more powerful than I know.

---

#### 4.2.2.6. Recovering partitions inside an extended partition

Extended partition information is scattered on the disk, not stored with the primary partition. To recover these often requires more work. The process is:

1. Scan for the start of the first partition (using gpart's -k option);
2. Create a temporary primary partition entry with the true start position and a fake end position. (This may drive you to delete an actual primary partition if no one is available - this is risk free if you don't reuse the sectors of the deleted partition);
3. Use "e2fsk -n", "mount -r", and "df" to determine the true end point. Write this value down (warning: read the man page for each program mentioned, and use the read-only options; you do not want to write to your disk until all partitions are in the correct place);
4. Repeat this process for each partition to be recovered;
5. Build a complete new correct partition table.

---

#### 4.2.2.7. If your hard drive has errors

If your hard drive has errors, you may have real trouble mounting, checking, or using data. (The drive read errors get in the way.) Gpart may not even find it. But if you know the start of the partition, you can easily copy the data to a temporary file stored on a different drive. Sectors with read errors will usually be set to zero

by this process:

- Copy the partition data to a file. You must know the start block of the partition;

```
dd if=/dev/hd?? of=/tmp/recover_hd?? bs=512 skip=XXXX count=YYY
```

XXX is the sector start and YYY the sector count (can be guessed).

- Mount the file as a loop file system.

```
mount -r -t ext2 -o loop /tmp/recover_hd?? /mnt/recover
```

Use dd_rescue if the disk is really badly damaged.

## 4.3. The rich man's case

Partition Magic is a commercial (and proprietary) software product, not so cheap given the little use one can have (approx a hundred bucks in France), but with a very high reputation all around there. However, I never use it and will not rate it. It's said to be able to do anything with partitions, including restoring them.

Ralf's original partition-rescue mini HOWTO was essentially based around the use of Partition Magic, so I assume it's a very good solution, if you have valuable data on your Linux partition and little Linux capability. However, there are now much more recent releases of Partition Magic and I think it's better for you to read the manual.

## 4.4. PMagic

PartedMagic if the tool of choice for any partition work, including recovery. Extremely good product (and open, of course). Read the HOWTO anyway, because you may need it to prevent disasters, but all the tools are on pmagic, this is the best recovery cd ever...

# 5. References

## 5.1. Authors

The author of this HOWTO is Jean-Daniel Dodin. I can be joined at My masqued e-mail or simply searching "Jean-Daniel Dodin" on Google :-).

My Web site is at [http://www.dodin.net].

I want to thank Rolf Klausen, who wrote the previous partition-rescue mini HOWTO. Even if I rewrote it almost entirely, he had first the good idea.

Every other member of the Linux community, and everybody who supports Linux and writes documentation and programs for Linux, and all the authors of the LDP, and virtually any person involved in anything which has to do with Linux. Particularly Linus B. Torvalds - he is **The King** !!!

I want also to thank Michail Brzitwa (see Web site in the text) for writing gpart !

Bryce Nesbitt <bryce at obviously dot com> did a very good job, "Linux's own info" is from him as are some minor enhancements.

---

## 5.2. Most recent version

The most recent version of this HOWTO will be found on the tldp wiki