

This is Google's cache of <https://gparted.org/h2-fix-msdos-pt.php>. It is a snapshot of the page as it appeared on Feb 11, 2018 12:08:31 GMT. The [current page](#) could have changed in the meantime. [Learn more](#)

[Full version](#)   [Text-only version](#)   [View source](#)

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac) and use the find bar.



## GNOME Partition Editor

- [Home](#)
- [Download](#)
- [Live CD/USB](#)
- [Help](#)
- [News](#)
- [Develop](#)
- [Donate](#)

## 1 CleanMyMac 3 - Get more out of your Mac.

Clean Up Your Mac OS X Right Now. Get Rid of Unneeded Files and Junk! [macpaw.com/CleanMyMac3](http://macpaw.com/CleanMyMac3)



## 2 Download Cleaner for Mac

Clean Mac files Right Now. Award-winning System Utility. [mackeeper.com](http://mackeeper.com)



# How-to Fix Invalid MSDOS Partition Tables

## Summary

This article describes how to fix some common problems that cause an MSDOS partition table (also known as Master Boot Record - MBR) to be invalid.

Note that a computer with an invalid partition table can often still boot up. However, tools like GParted will not work until the partition table is corrected.

**CAUTION:** Editing partitions has the potential to cause **LOSS of DATA**. You are advised to **BACKUP your DATA** before proceeding.

## Contents

- [Basic MSDOS Partition Table Rules](#)
- [Symptoms of an Invalid Partition Table](#)
  - [Entire Disk Device Shown as Unallocated](#)
  - [Applying Action on Logical or Extended Partition Fails](#)
- [Instructions \(step-by-step\)](#)
  - [How-to Fix Overlapping Partitions](#)
  - [How-to Fix Partition Outside the Disk](#)
  - [How-to Fix Unable to Satisfy All Constraints](#)

## MSDOS Partition Table Rules

**TIP:** Primary and extended partitions are identified by numbers 1 to 4 (e.g., /dev/sdb3). Logical partitions are identified by numbers 5 and higher (e.g., /dev/sdb7).

A valid MSDOS partition table must obey all of the following rules:

1. At most four primary partitions, or three primary partitions and one extended partition are permitted.
2. Primary and Extended partitions must not overlap.
3. Primary partitions must not exist within an Extended partition.
4. An extended partition may contain multiple logical partitions.
 

It is permissible for a logical partition to end on the same sector as the end of the extended partition.
5. Logical partitions must not overlap other logical partitions.
6. At least 2 unallocated sectors are required in front of a logical partition.
 

This space is required for the Extended Boot Record (EBR).
7. All partitions must exist within the size of the disk device.
 

Sector numbering begins at zero, so the first sector is sector 0.
8. Each partition must start within the first  $2^{32}$  (4,294,967,296) sectors of the disk device, and be smaller than  $2^{32}$  sectors.
 

For disks with 512 byte sectors,  $2^{32}$  sectors is 2 Tebibytes (2 TiB).

## Symptoms of an Invalid Partition Table

You might have an invalid partition table if you notice either of the following symptoms when using GParted:

- [GParted shows the entire disk device as unallocated](#)

- [GParted fails when applying an action on a logical or an extended partition](#)

## Entire Disk Device Shown as Unallocated

There are a few situations that can cause this problem. To learn the cause, in GParted double-click on the "unallocated" disk device and check the information window for warning messages.

Common warning messages that cause this problem are as follows:

- **Can't have overlapping partitions**

The root cause is that the partition table contains at least one occurrence where the end of one partition overlaps the start of another partition. See [How-to Fix Overlapping Partitions](#).

- **Can't have a partition outside the disk!**

The root cause is that the partition table contains at least one occurrence where the end of one partition is beyond the end of the disk device. See [How-to Fix Partition Outside the Disk](#).

- **/dev/sda: unrecognized disk label**

Two root causes for this message are as follows:

- **The device does not contain a partition table.**

If the device does contain data then perhaps the entire device is formatted with a file system. It is also possible that the device is part of some other data structure, such as a RAID. In these situations there is no partition table to edit.

Otherwise, if this is a new disk device then the device is most likely empty. In this case you would proceed to create a partition table and at least one partition to prepare the disk device for use.

- **The type of partition table is not recognized by GParted.**

If the device contains data, then perhaps the device is using a new type of partition table. GParted supports many commonly used partition tables, such as msdos or gpt, but not all types of partition tables are supported. In this situation you would need to determine what type of partition table is in use. Then you could seek out other partition management tools to edit the partition table.

## Applying Action on Logical or Extended Partition Fails

To determine if the failure is due to an invalid partition table, expand the GParted details in the "Applying pending operations" window, and search for the source of the error.

If the text "**Unable to satisfy all constraints on the partition**" is found, then there might be less than 2 unallocated sectors in front of each logical partition.

For example, there should be at least 2 unallocated sectors between:

- the start of the extended partition and the start of a logical partition.
- the end of one logical partition and the start of the next logical partition.

The 2 unallocated sectors are required to store the Extended Boot Record (EBR). The EBR contains information about the following logical partition. The Linux kernel considers EBRs to be two sectors long, in order to "leave room for LILO (Linux LOader)".

See [How-to Fix Unable to Satisfy All Constraints](#).

## Instructions (step-by-step)

**TIP:** The **sudo** prefix is used to acquire root privilege.

### How-to Fix Overlapping Partitions

The following instructions describe how to manually correct the problem of overlapping partitions.

Other methods also exist, such as using [testdisk](#) to scan the disk device to rebuild the partition table. The testdisk application is included on [GParted Live](#).

**NOTE:** Be sure to choose the correct disk device path.

In the following example, the disk device containing overlapping partitions is **/dev/sda**

1. Ensure you have a good backup of your data.
2. Confirm the problem by running parted on your disk device (e.g., **/dev/sda**).

For example:

```
$ sudo parted /dev/sda unit s print
Error: Can't have overlapping partitions.
```

You should see the error message *Error: Can't have overlapping partitions.*

### 3. Gather partition details for analysis using the fdisk command.

For example:

```
$ sudo fdisk -l -u /dev/sda
```

```
Disk /dev/sda: 60.0 GB, 60011642880 bytes
255 heads, 63 sectors/track, 7296 cylinders, total 117210240 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000b2d37

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           63      81922347   40961142+    7  HPFS/NTFS/exFAT
/dev/sda2             81915435   117210239   17647402+    5  Extended
/dev/sda5             81915498   98494514    8289508+   83  Linux
/dev/sda6             98494578   99008594     257008+   82  Linux swap / Solaris
/dev/sda7             99008658   115394894    8193118+    b  W95 FAT32
```

### 4. Check the fdisk output for the following problems:

- *Do any of the primary or extended partitions overlap?*

In other words does one partition (numbers 1 to 4) have a start value lower than another partition (numbers 1 to 4) but also an end value higher than the start of the other partition?

- *Do any of the primary partitions exist within the extended partition?*

In other words does one partition (numbers 1 to 4) have start and end values within the boundaries of another partition (number 1 to 4)? If this is the situation then either the extended partition must be altered so that it does not encompass the primary partition, or the primary partition must be changed to be a logical partition within the extended partition. Fixing this situation might require backing up the partitions, repartitioning the disk, restoring the partition contents, and then repairing the ability to boot.

- *Do any of the logical partitions start or end outside the extended partition?*

In other words is the start of a logical partition (numbers 5+) less than the start of the extended partition? Also, is the end of a logical partition (numbers 5+) greater than the end of the extended partition?

- *Do any of the logical partitions overlap?*

In other words does one logical partition (numbers 5+) have a start value lower than another logical partition (numbers 5+) but also have an end value higher than the start of the other partition?

In this example the end of primary partition sda1 overlaps the beginning of extended partition sda2.

```
81,922,347 --- end of sda1
81,915,435 --- start of sda2
```

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           63      81922347   40961142+    7  HPFS/NTFS/exFAT
/dev/sda2             81915435   117210239   17647402+    5  Extended
/dev/sda5             81915498   98494514    8289508+   83  Linux
/dev/sda6             98494578   99008594     257008+   82  Linux swap / Solaris
/dev/sda7             99008658   115394894    8193118+    b  W95 FAT32
```

### 5. Determine a new end value (and partition size) to avoid the overlap.

#### NOTE:

- In cases where we do not know if the starting or ending sector is the problem, we assume that the starting sector of each partition is correct, and that the ending sector might be in error.
- No sectors are required between primary and extended partitions.
- At least 2 sectors of unallocated space are required in front of logical partitions.

In this example, we assume that the starting sector of extended partition sda2 is correct. Hence we need to change the end sector of primary partition sda1.

The new end sector of sda1 should be the start of sda2 minus one sector.

\* *For logical partitions we would need to subtract at least two (2) sectors.*

```
new sda1 end = (sda2 start) - 1
              = 81915435 - 1
              = 81915434
```

New size of sda1 will be the new end of sda1 minus the start of sda1 plus one sector.

```
new sda1 size = (new sda1 end) - (sda1 start) + 1
              = 81915434 - 63 + 1
              = 81915372
```

6. Make a copy of the partition table in an editable file using the `sfdisk` command.

For example:

```
$ sudo sfdisk -d /dev/sda > sda-backup.txt
```

7. Use your favourite editor to edit the file to change the old partition size to the new partition size.  
If you are using GParted Live, you can edit the file using the Leafpad editor.

For example:

```
$ sudo leafpad sda-backup.txt

# partition table of /dev/sda
unit: sectors

/dev/sda1 : start=      63, size= 81922285, Id= 7, bootable
/dev/sda2 : start= 81915435, size= 35294805, Id= 5
/dev/sda3 : start=      0, size=      0, Id= 0
/dev/sda4 : start=      0, size=      0, Id= 0
/dev/sda5 : start= 81915498, size= 16579017, Id=83
/dev/sda6 : start= 98494578, size=   514017, Id=82
/dev/sda7 : start= 99008658, size= 16386237, Id= b
```

Change the old size of the primary partition `sda1` (81922285) to the calculated new size (81915372).

```
# partition table of /dev/sda
unit: sectors

/dev/sda1 : start=      63, size= 81915372, Id= 7, bootable
/dev/sda2 : start= 81915435, size= 35294805, Id= 5
/dev/sda3 : start=      0, size=      0, Id= 0
/dev/sda4 : start=      0, size=      0, Id= 0
/dev/sda5 : start= 81915498, size= 16579017, Id=83
/dev/sda6 : start= 98494578, size=   514017, Id=82
/dev/sda7 : start= 99008658, size= 16386237, Id= b
```

Save the file and exit the editor.

8. Write the corrected partition details to the partition table using the `sfdisk` command.

For example:

```
$ sudo sfdisk /dev/sda < sda-backup.txt
Checking that no-one is using this disk right now ...
OK

Disk /dev/sda: 7296 cylinders, 255 heads, 63 sectors/track
Old situation:
Units = cylinders of 8225280 bytes, blocks of 1024 bytes, counting from 0

   Device Boot  Start      End  #cyls   #blocks  Id System
/dev/sda1  *           0+    5099-   5100-   40961142+   7 HPFS/NTFS/exFAT
/dev/sda2             5099    7295   2197   17647402+   5 Extended
/dev/sda3              0         -         0         0   0 Empty
/dev/sda4              0         -         0         0   0 Empty
/dev/sda5             5099+    6130   1032-   8289508+   83 Linux
/dev/sda6             6131+    6162     32-   257008+   82 Linux swap / Solaris
/dev/sda7             6163+    7182   1020-   8193118+   b  W95 FAT32
New situation:
Units = sectors of 512 bytes, counting from 0

   Device Boot  Start      End  #sectors  Id System
/dev/sda1  *           63   81915434   81915372   7 HPFS/NTFS/exFAT
/dev/sda2      81915435 117210239   35294805   5 Extended
/dev/sda3              0         -         0   0 Empty
/dev/sda4              0         -         0   0 Empty
/dev/sda5      81915498 98494514   16579017   83 Linux
/dev/sda6      98494578 99008594    514017   82 Linux swap / Solaris
/dev/sda7     99008658 115394894   16386237   b  W95 FAT32
Successfully wrote the new partition table

Re-reading the partition table ...

If you created or changed a DOS partition, /dev/foo7, say, then use dd(1)
to zero the first 512 bytes: dd if=/dev/zero of=/dev/foo7 bs=512 count=1
(See fdisk(8).)
```

9. Confirm the problem is now resolved by running `parted` on your disk device.

For example:

```
$ sudo parted /dev/sda unit s print
Model: ATA ST3060022ACE (scsi)
Disk /dev/sda: 117210240s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start      End          Size         Type         File system  Flags
```

1	63s	81915434s	81915372s	primary	ntfs	boot
2	81915435s	117210239s	35294805s	extended		
5	81915498s	98494514s	16579017s	logical	ext2	
6	98494578s	99008594s	514017s	logical	linux-swaps(v1)	
7	99008658s	115394894s	16386237s	logical	fat32	

You should now be able to use GParted with this device. :-)

## How-to Fix Partition Outside the Disk

The following instructions describe how to manually correct the problem of a partition extending beyond the end of the disk.

Other methods also exist, such as using [testdisk](#) to scan the disk device to rebuild the partition table. The testdisk application is included on [GParted Live](#).

If the problem occurs with the extended partition, then you might consider using the [fixparts](#) application. The fixparts application is also included on [GParted Live](#).

**NOTE:** Be sure to choose the correct disk device path.

In the following example, the disk device containing a partition outside the disk is **/dev/sdb**

1. Ensure you have a good backup of your data.
2. Confirm the problem by running parted on your disk device (e.g., */dev/sdb*).

For example:

```
$ sudo parted /dev/sdb unit s print
Error: Can't have a partition outside the disk!
```

You should see the error message *Error: Can't have a partition outside the disk!*

3. Gather partition details for analysis using the fdisk command.

For example:

```
$ sudo fdisk -l -u /dev/sdb

Disk /dev/sdb: 250.1 GB, 250059350016 bytes
255 heads, 63 sectors/track, 30401 cylinders, total 488397168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00068df3

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *          63       12578894       6289416    7  HPFS/NTFS/exFAT
/dev/sdb2             12578895      180345689      83883397+    7  HPFS/NTFS/exFAT
/dev/sdb3          180345690      390058199      104856255    7  HPFS/NTFS/exFAT
/dev/sdb4          390058200      488408189       49174995    5  Extended
/dev/sdb5          390058263      459089504       34515621   83  Linux
/dev/sdb6          459089568      488392064       14651248+   82  Linux swap / Solaris
```

4. Check the fdisk output for the cause of the problem.

*Does any partition have an end value larger than the disk size?*

To be precise, since the first sector begins at sector zero (0), check to see if the end of any partition is greater than the disk size minus one sector.

In this example the end of extended partition sdb4 is beyond the size of the disk.

488,408,189 --- end of sdb4

488,397,168 --- size of sdb device

```
Disk /dev/sdb: 250.1 GB, 250059350016 bytes
255 heads, 63 sectors/track, 30401 cylinders, total 488397168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00068df3

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *          63       12578894       6289416    7  HPFS/NTFS/exFAT
/dev/sdb2             12578895      180345689      83883397+    7  HPFS/NTFS/exFAT
/dev/sdb3          180345690      390058199      104856255    7  HPFS/NTFS/exFAT
/dev/sdb4          390058200      488408189       49174995    5  Extended
/dev/sdb5          390058263      459089504       34515621   83  Linux
/dev/sdb6          459089568      488392064       14651248+   82  Linux swap / Solaris
```

5. Determine a new end value (and partition size) to prevent the partition outside the disk problem.

The new end sector of the partition outside the disk should be the size of the disk minus one sector.

In our example:

```
new sdb4 end = (sdb disk size) - 1
             = 488397168 - 1
             = 488397167
```

New size of sda1 will be the new end of sda1 minus the start of sda1 plus one sector.

```
new sdb4 size = (new sdb4 end) - (sdb4 start) + 1
              = 488397167 - 390058200 + 1
              = 98338968
```

6. Make a copy of the partition table in an editable file using the `sfdisk` command.

For example:

```
$ sudo sfdisk -d /dev/sdb > sdb-backup.txt
```

7. Use your favourite editor to edit the file to change the old partition size to the new partition size.

If you are using GParted Live, you can edit the file using the Leafpad editor.

For example:

```
$ sudo leafpad sdb-backup.txt

# partition table of /dev/sdb
unit: sectors

/dev/sdb1 : start=      63, size= 12578832, Id= 7, bootable
/dev/sdb2 : start= 12578895, size=167766795, Id= 7
/dev/sdb3 : start=180345690, size=209712510, Id= 7
/dev/sdb4 : start=390058200, size= 98349990, Id= 5
/dev/sdb5 : start=390058263, size= 69031242, Id=83
/dev/sdb6 : start=459089568, size= 29302497, Id=82
```

Change the old size of the partition sdb4 (98349990) to the calculated new size (98338968).

```
# partition table of /dev/sdb
unit: sectors

/dev/sdb1 : start=      63, size= 12578832, Id= 7, bootable
/dev/sdb2 : start= 12578895, size=167766795, Id= 7
/dev/sdb3 : start=180345690, size=209712510, Id= 7
/dev/sdb4 : start=390058200, size= 98338968, Id= 5
/dev/sdb5 : start=390058263, size= 69031242, Id=83
/dev/sdb6 : start=459089568, size= 29302497, Id=82
```

Save the file and exit the editor.

8. Write the corrected partition details to the partition table using the `sfdisk` command.

For example:

```
$ sudo sfdisk /dev/sdb < sdb-backup.txt
Checking that no-one is using this disk right now ...
OK

Disk /dev/sdb: 30401 cylinders, 255 heads, 63 sectors/track
Old situation:
Units = cylinders of 8225280 bytes, blocks of 1024 bytes, counting from 0

   Device Boot Start      End  #cyls   #blocks  Id System
/dev/sdb1  *           0+       782     783-    6289416   7 HPFS/NTFS/exFAT
/dev/sdb2           783    11225   10443   83883397+   7 HPFS/NTFS/exFAT
/dev/sdb3        11226    24279   13054   104856255   7 HPFS/NTFS/exFAT
/dev/sdb4        24280   30402-    6123-   49174995   5 Extended
/dev/sdb5        24280+   28576    4297-   34515621   83 Linux
/dev/sdb6        28577+   30400    1824-   14651248+   82 Linux swap / Solaris
New situation:
Units = sectors of 512 bytes, counting from 0

   Device Boot  Start      End  #sectors  Id System
/dev/sdb1  *         63   12578894   12578832   7 HPFS/NTFS/exFAT
/dev/sdb2    12578895  180345689   167766795   7 HPFS/NTFS/exFAT
/dev/sdb3   180345690  390058199   209712510   7 HPFS/NTFS/exFAT
/dev/sdb4   390058200  488397167    98338968   5 Extended
/dev/sdb5   390058263  459089504    69031242   83 Linux
/dev/sdb6   459089568  488392064    29302497   82 Linux swap / Solaris
Warning: partition 4 does not end at a cylinder boundary
Warning: partition [6] does not end at a cylinder boundary
Successfully wrote the new partition table
```

Re-reading the partition table ...

If you created or changed a DOS partition, `/dev/foo7`, say, then use `dd(1)` to zero the first 512 bytes: `dd if=/dev/zero of=/dev/foo7 bs=512 count=1`

(See fdisk(8).)

9. Confirm the problem is now resolved by running parted on your disk device.

For example:

```
$ sudo parted /dev/sdb unit s print
Model: ATA ST3250022ACE (scsi)
Disk /dev/sdb: 488397168s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start      End          Size         Type         File system  Flags
  1      63s        12578894s    12578832s    primary      ntfs         boot
  2      12578895s  180345689s   167766795s   primary      ntfs
  3      180345690s 390058199s   209712510s   primary      ntfs
  4      390058200s 488397167s   98338968s    extended
  5      390058263s 459089504s   69031242s    logical      ext2
  6      459089568s 488392064s   29302497s    logical      linux-swap(v1)
```

You should now be able to use GParted with this device. :-)

How-to Fix Unable to Satisfy All Constraints

We plan to outline how to approach this problem. In the meantime you can seek help in the [GParted forum](#).

