Engineering White Paper

EMC² where information lives®

# Using diskpar and diskpart to Align Partitions on Windows Basic and Dynamic Disks

***Abstract***

This white paper describes how to use `diskpar` and `diskpart` to align partitions on Basic Disks and Dynamic Disks under both Windows 2000 and Windows 2003. This paper provides extensive analysis to demonstrate how `diskpar` works and the limitations of using it.

Published September 2005

Part Number H1445

# Table of Contents

# Executive Summary

This white paper describes how to use `diskpar` and `diskpart` to align partitions on Basic Disks and Dynamic Disks under both Windows 2000 and Windows 2003. This paper provides extensive analysis to demonstrate how `diskpar` works and the limitations of using it.

Note that each section is separate in its own right and hence may appear to repeat information in a previous section. This is intentional.

# The Need for Partition Alignment

To maximize disk performance, any I/O to an EMC® Symmetrix® or EMC CLARiiON® array needs to be structured to prevent any single I/O operation "straddling" (crossing) any "significant" boundaries in the EMC Storage. If an I/O does straddle a boundary, this can consume extra resources or cause additional work in the storage array leading to performance loss. There are three significant boundaries:

- Symmetrix Cache Slot Boundaries (one Track [64 Blocks] – 32 KB)
- RAID 5 Boundaries (four Tracks [256 Blocks] – 128 KB)
- Metastripe Boundaries (two Cylinders [1920 Blocks] – 983 KB)

We want to try to minimize the possibility of any single I/O causing a write to *both* sides of any of the above boundaries.

Owing to the legacy of the IBM PC BIOS, Windows disks over 7.8 GB are usually deemed to have a geometry of 63 sectors/track and 255 heads/cylinder. Note that sector numbers always start with one (not zero). This means that the first sector on a track is sector one, *not* sector zero. Additionally, the next track begins with sector one again. However, it is far more convenient to think of a disk as a sequence of blocks starting from address zero and incrementing until the end of the disk. Because of this, it is important to think in terms of blocks rather than sectors. As an aside, when Windows interrogates a disk for the drive geometry, only one byte is reserved for the number of heads. This means that the maximum number of heads that can be reported is 255. Head addresses always start at zero, which means that the last head number is 254.

With SCSI disks, unless the SCSI BIOS is enabled, Windows cannot determine the geometry of the disk. The only information it has access to is the number of blocks on the disk. Windows therefore assumes that the disk geometry is as above (63 sectors/track and 255 heads/cylinder). Windows takes the block count from the disk and calculates the number of cylinders. To determine the number of cylinders, Windows performs the following calculation (Microsoft KB323231):

$$Cylinders = \text{int}\left(\frac{Blocks}{63 * 255}\right)$$

By default, partitions created on disks are normally aligned on a cylinder (as defined by Windows) boundary, with one exception. The first partition after the MBR (Master Boot Record) is actually track-aligned, presumably since it was felt that it was too wasteful to just have one block (the MBR) in an empty cylinder. This is a legacy issue.

`Diskpar` and `diskpart` allow the creation of a primary partition at any desired block address rather than the default 63 blocks. This means that a partition can be created to minimize the boundary crossings mentioned earlier.

Block 0 on the disk contains the MBR, which defines the disk layout. Remembering that partitions are created on cylinder boundaries, the first partition cannot be created on top of the MBR; therefore, the partition is created at the next track boundary. This is at block address 63. Remember that you start counting blocks at zero, not one, so the first partition starts at the 64th block (and stating the obvious—there are 63 blocks before this first partition). The problem is that EMC Symmetrix storage defines tracks differently. On a Symmetrix array, a track is considered to be 64 blocks and a Symmetrix cache memory

slot is based on this track size and offset. On EMC CLARiiON storage, the unit of allocation is an element, which is (by default) 128 KB.

## Track Boundaries

If we use the Windows default partition location (63), an I/O of 4 KB (eight blocks) starting at the beginning of the partition will write one block to the last block of the first Symmetrix track and seven blocks to the start of the second Symmetrix track. This means the I/O has straddled the first and second Symmetrix tracks. This requires the Symmetrix array to reserve two cache slots for the data and also requires two flush I/O operations to the Symmetrix disk, which impacts performance.

For I/O to this partition:

- Any I/O of 32 KB or larger will always cause a boundary crossing.
- Any random I/O of 16 KB will cause a boundary crossing 50 percent of the time.
- Any random I/O of 8 KB will cause a boundary crossing 25 percent of the time.
- Any random I/O of 4 KB will cause a boundary crossing 12.5 percent of the time.

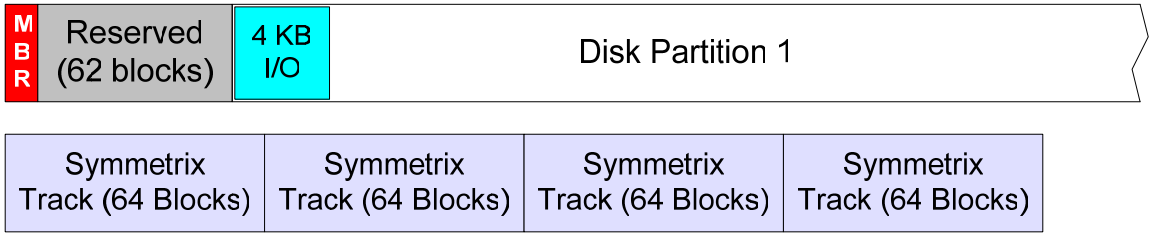



**Figure 1. Misaligned Partition (Not to Scale)**

As Figure 1 shows, by default the first partition starts on block address 63, whereas to be aligned with a Symmetrix track, it should start at block address 64. A 4 KB I/O at the start of the disk partition will cause two cache memory slots to be reserved (one for each track).

If the partition started at block 64 (zero based), then no I/O (of 32 KB or less) would cause any boundary crossings.

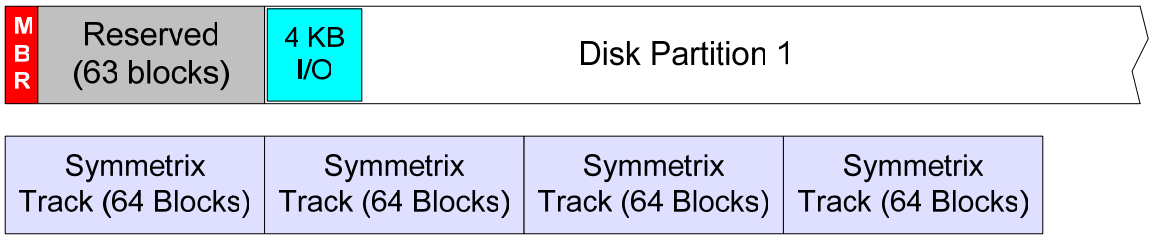



**Figure 2. Aligned Partition (Not to Scale)**

As Figure 2 shows, starting the first partition on block address 64 will align the I/Os and will not cause boundary crossings.

## RAID 5 Boundaries

If we now consider RAID 5 (DMX series) volumes, things change. There are two differences to consider. First, the maximum I/O size that Windows will issue is 64 KB. Larger I/Os are broken up into 64 KB chunks. (*Windows 2000 Server Operations Guide*, Chapter 8, I/O Request size). Second, the RAID 5 stripe size is four Symmetrix tracks (256 blocks).

If the partition is aligned at 64 blocks (we found that this was a good value earlier), we will *still* have a potential performance problem. Assume that an I/O write of 64 KB (132 blocks) is issued to the start of the

partition. This will not be a problem. Two cache memory slots are required but they are the minimum required for this I/O size. If another I/O (sequential write) of 64 KB is issued, then there is a problem. This second I/O straddles two RAID 5 stripe elements and requires the two stripes to be updated. This requires twice the resources at the back end of the Symmetrix array compared with an I/O that does not cross a stripe boundary.

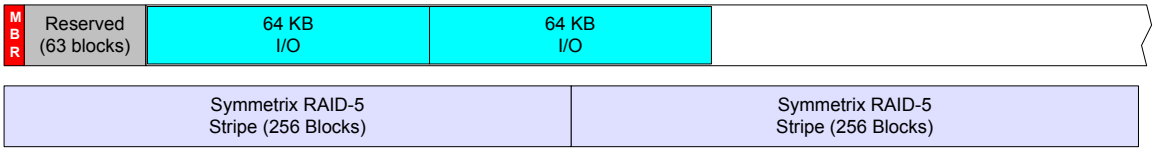



**Figure 3. Misaligned RAID 5 Stripe (Not to Scale)**

As Figure 3 shows, starting the first partition on block address 64 will cause stripe boundary crossings for 64 KB I/Os. To correct this, the partition needs to be aligned to 128 blocks.

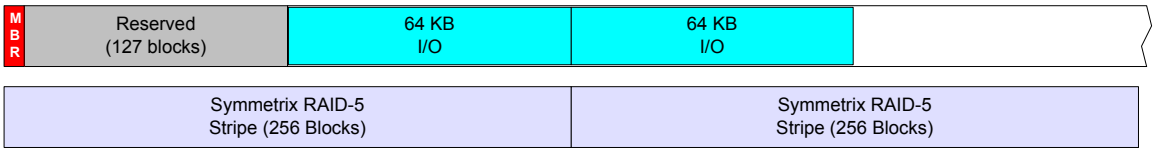



**Figure 4. Aligned RAID 5 Stripe (Not to Scale)**

As Figure 4 shows, starting the first partition on block address 64 will not cause stripe boundary crossings for 64 KB I/Os.

## *Metastripe Boundaries*

Metastripes are two Symmetrix cylinders (30 Symmetrix tracks) in size. Metastripe boundaries have the same back-end performance hit as RAID 5 stripe but have one further complication when EMC SRDF® is used. If an I/O crosses a metastripe boundary, it is broken up into two I/O operations for RDF purposes (one for either side of the boundary). In RDF Journal Mode 0, this means that the acknowledgement for the first I/O must be received before the second I/O can be sent. In a similar fashion, if the data is being fetched from the R2 side while performing a RDF restore, the I/O will be broken up into two RDF I/O reads.

Given that the Windows maximum I/O transfer size is 64 KB (128 blocks) and the Metastripe size is 30 Symmetrix cylinders (1920 blocks), we can see that an alignment of 128 will work since 128 is a multiple of 1920: (1920 / 128 = 15). If Windows ever permits I/O of 128 KB, we would then have a problem since we would need to align the partition based on the lowest common multiple of 128 KB (256 blocks) and 1920 blocks, which is 3840 blocks (four Symmetrix cylinders).

## *EMC CLARiiON Arrays*

On a CLARiiON array, the default element size is 128 blocks, but it can be from four to 256 blocks.

## Determining the Correct Offset to Partition

You must first determine where to start the partition:

- For a Symmetrix array, the start address is at block 128.
- For a CLARiiON array, you must determine the element size. To do this, you can use EMC Navisphere® Array Manager.

To determine the element size, follow these steps:

1. Start Navisphere.

2. Navigate to the appropriate storage group and LUN to be used.

   For example, suppose you want to create an aligned partition on LUN 2 on the lab-w2k host. Note that the LUN number you see in this display is *not* the LUN number that the host sees. The LUN number in the display is the array LUN number. To get the host LUN number, you must look at the **Storage** tab in the host properties.

3. Right-click the LUN and select **Properties**.

   Figure 5 shows the **LUN Properties** dialog box.

4. Ensure that the **Alignment Offset** value is 0. If not, the CLARiiON LUN has been deliberately misaligned, possibly to correct the partition misalignment that you intend to correct. If this is the case, the **Alignment Offset** value would most likely be 63. If a value other than 0 or 63 appears in the field, then further investigation is required to determine the reason for having a nonstandard value.

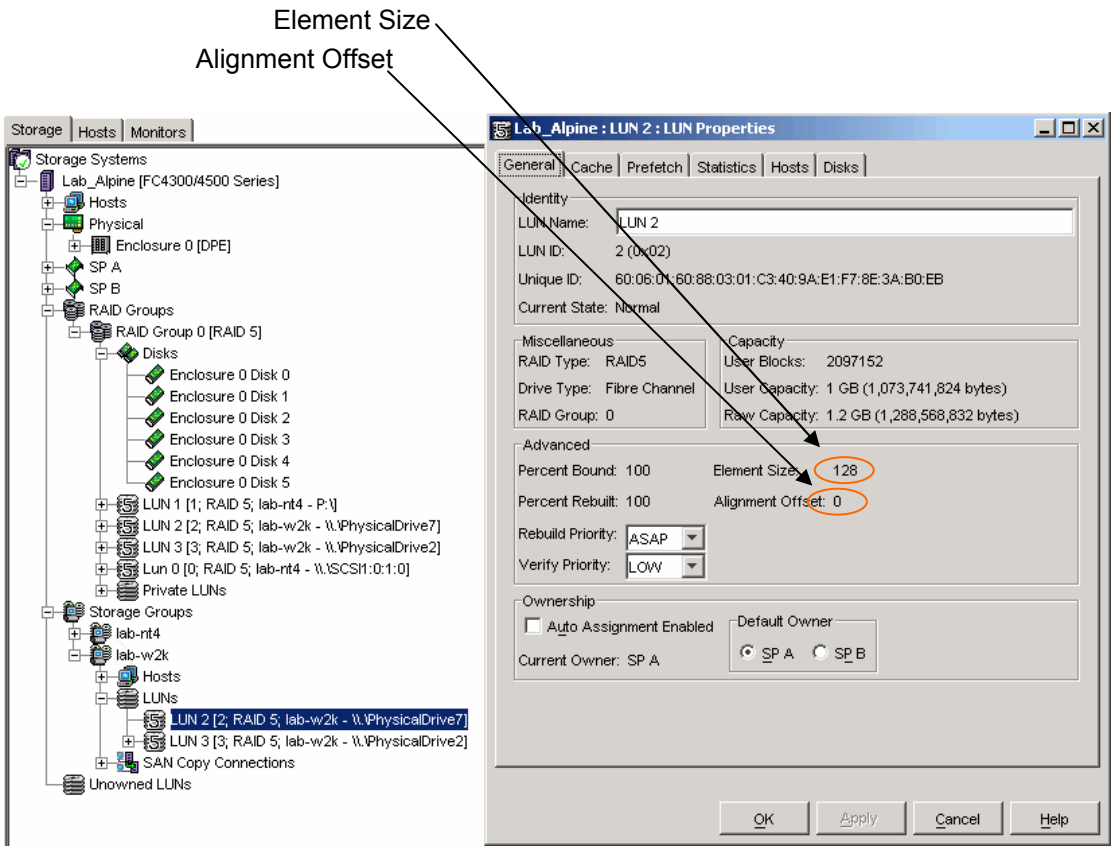



**Figure 5. LUN Properties**

## *Diskpar and diskpart*

There are two programs available for aligning partitions. The `diskpar` utility has been around for some time and works on all Windows Systems. The `diskpart` utility has been available since Windows 2000, but could not align partitions until an updated version was included with Windows 2003 SP1. The version of `diskpart` that can align partitions is version 5.2.3790 or higher.

To summarize:

| | diskpar | diskpart (5.2.3790) |
|---|---|---|
| Windows NT | ✓ | |
| Windows 2000 | ✓ | |
| Windows 2003 | ✓ | |
| Windows 2003 SP1 | ✓ | ✓ |

## Running diskpar

Refer to the following sections for details on running `diskpar`:


- *Using diskpar with Windows 2000 Basic Disks* on page 11
- *Using diskpar with Windows 2003 Basic Disks* on page 17
- *Using diskpart with Windows 2003 SP1 Basic Disks* on page 23
- *Using diskpar with Windows 2000 Dynamic Disks* on page 29
- *Using diskpar with Windows 2003 Dynamic Disks* on page 39
- *Using diskpart with Windows 2003 SP1 Dynamic Disks* on page 47

# Using diskpar with Windows 2000 Basic Disks

Starting with a completely empty Symmetrix volume of 18414 cylinders, the following sections present some interesting blocks as the steps are performed. Use the `secinspect` tool from the Windows 2000/2003 Resource Kit.

| Block | Contents |
|---|---|
| Block 0 | Master Boot Record. |
| Block 63 | Default block for the first partition created. This is aligned on a track boundary where the definition of a track is 63 blocks. |
| Block 128 | Block to use for the new partition because it is aligned on a 128-block boundary. If this was a CLARiiON array, you would most likely be looking at block 128. |

## *Step 1: Write the Disk Signature*

After the signature has been written, the MBR contains the following:

**Block 0 – Master Boot Record**

```
LBN 0

******
0x01b0                          -a9 a0 94 a5 00 00 00 00
0x01c0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This shows a valid signature (0xA594A0A9) and no partitions defined.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

**Block 128 – The (aligned) block address on which to create a partition**

```
LBN 128

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

## *Step 2: Run diskpar and Create a Partition*

This example shows how to run `diskpar` on PhysicalDrive1 and create a 4 GB partition.

```
C:\ diskpar -s 1
Set partition can only be done on a raw drive.
You can use Disk Manager to delete all existing partitions
Are you sure drive 1 is a raw device without any partition? (Y/N) y

---- Drive 1 Geometry Information ----
```

```
Cylinders = 1174
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 9656478720 (Bytes) = 9209 (MB)

We are going to set the new disk partition.
All data on this drive will be lost. continue (Y/N)? Y

Please specify starting offset (in sectors): 128
Please specify partition length (in MB) (Max = 9209): 4000

Done setting partition.
---- New Partition information ----
StatringOffset[1] = 65536
PartitionLength = 4194304000
HiddenSectors = 128
PartitionNumber = 1
PartitionType = 7
```

You should now use Disk Manager to format this partition.

Note that, unlike Windows 2003, the Windows 2000 Disk Management GUI does *not* update automatically with the new partition information. If the Disk Management GUI was open when `diskpar` was run, you must restart the GUI.

## *Step 3: Format the Partition*

After formatting, the MBR contains the following:

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-a9 a0 94 a5 00 00 00 01
0x01c0   03 00 07 ee 71 fd 80 00-00 00 00 00 7d 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The offset to the partition is 0x00000080 (128), which is what you specified in `diskpar`.

The length of the partition is 0x007D0000 (8,192,000) blocks (4 GB).

(For information on the structure of the MBR, refer to Appendix A.)

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

The zeroes indicate that the block has not been touched. The newly created partition definitely does not start here.

**Block 128 – The (aligned) block address where you created the 4 GB partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff ff 7c 00 00 00 00 00   ....Ç.Ç.  |.....
```

[1] The pedants amongst us will argue for hours whether to correct the spelling from the `diskpar` output.

```
0x0030   04 00 00 00 00 00 00 00-ff cf 07 00 00 00 00 00   ........ -......
0x0040   f6 00 00 00 01 00 00 00-4d ec 80 7c 00 81 7c 46   ÷.......M8Ç|.ü|F
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

This is the NTFS boot record for the new volume. Note the value at offset 0x28. This is the size of the partition in blocks. The value is 0x007CFFFF (8,191,999). This is one less than the actual size you specified, and it ensures that the backup copy of the NTFS boot record (the last block in this partition) is untouched.

If we look at block 128 with DskProbe2 (using the NTFS… view), we see the following:

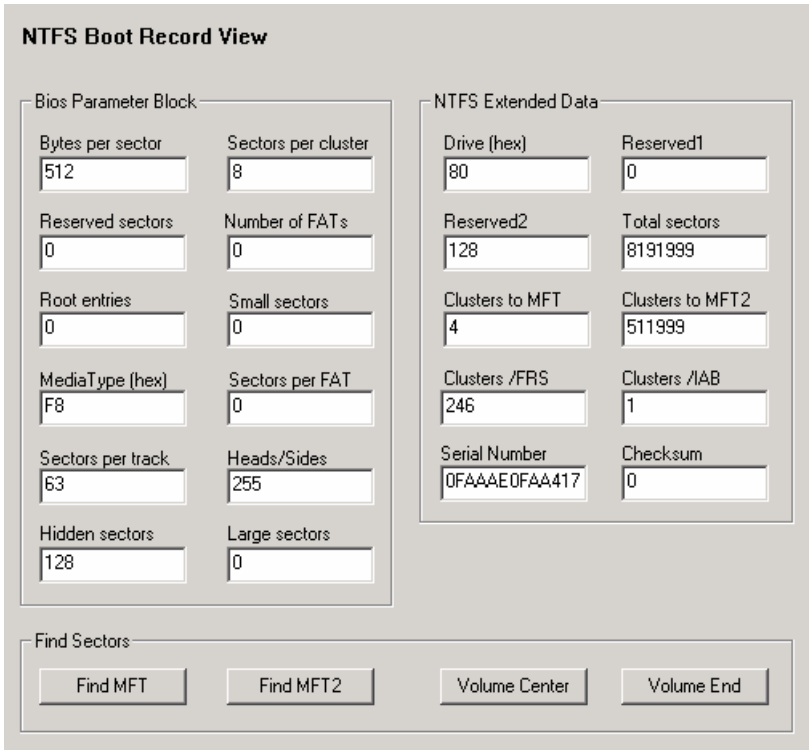



**Figure 6. Viewing the NTFS Boot Record**

This looks like a good NTFS boot record. The partition length checks out, but as a further check, look at the backup NTFS boot record located at the last block in the partition, which follows.

Offset to this partition + Size of partition – 1
(128) + (8,192,000) – 1 = 8,192,127

```
LBN 8192127

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff ff 7c 00 00 00 00 00   ....Ç.Ç.  |.....
0x0030   04 00 00 00 00 00 00 00-ff cf 07 00 00 00 00 00   ........ ¯......
0x0040   f6 00 00 00 01 00 00 00-4d ec 80 7c 00 81 7c 46   ÷.......M8Ç|.ü|F
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.ì+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

It is the same as the NTFS boot record at block 128. So this *is* the last block of the partition.

## Step 4: Create Another 4 GB Partition and Check the Alignment

You must use the LDM GUI to do this (or the `diskpart` utility). Do not use `diskpar` since this will destroy the first partition you just created.

Create another 4 GB partition from the LDM GUI.

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-a9 a0 94 a5 00 00 00 01
0x01c0   03 00 07 ee 71 fd 80 00-00 00 00 00 7d 00 00 ee
0x01d0   72 fd 07 fe ff fb 80 00-7d 00 7c 08 7d 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The new 4 GB partition is at offset 0x007D0080 (8,192128). This is aligned (exactly divisible by 128) so you now have two 4 GB partitions and they are both aligned. The length of this new partition looks a bit suspicious though. The length is 0x007D087C (8,194,172) and this is *not* exactly divisible by 128. By itself, this is not an issue since the only time a problem occurs is when accessing the short last segment of the partition, which is 124 blocks (8,194,172 MOD 128) instead of 128 blocks.

But what if you create another partition?

## Step 5: Create Another 1 GB Partition and Check the Alignment

Create another 1 GB partition from the LDM GUI.

```
LBN 0

0x01b0   00 00 00 00 00 2c 44 63-ae ce 15 f4 00 00 00 02
0x01c0   03 00 07 ee 71 fd 80 00-00 00 00 00 7d 00 00 ee
0x01d0   72 fd 07 fe ff fb 80 00-7d 00 7c 08 7d 00 00 00
0x01e0   c1 fc 06 fe ff ff fc 08-fa 00 bf 21 1f 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 55 aa
```

This third partition is at offset 0x00FA08FC (16,386,300). This offset is *not* divisible by 128. Interestingly though, it *is* divisible by 63.

As mentioned before, the Disk Manager GUI will align partitions on cylinder boundaries. It has aligned the end of the second partition on a cylinder boundary, which means that the third partition starts (and ends) on a cylinder boundary and is therefore *not* aligned.

## Step 6: Create the Second 4 GB Partition with diskpart and Check the Alignment

If you delete the last two partitions you just created (or run `diskpar` again), you can create a second 4 GB partition using `diskpart` instead of the Disk Manager GUI.

```
C:\ diskpart

Microsoft DiskPart version 1.0
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: VM-SRV1

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           4000 MB    64 KB

DISKPART> create partition primary size=4000

DiskPart succeeded in creating the specified partition.

DISKPART> list  partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           4000 MB    64 KB
* Partition 2    Primary           4001 MB  4001 MB

DISKPART>
```

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 2c 44 63-ae ce 15 f4 00 00 00 02
0x01c0   03 00 07 ee 71 fd 80 00-00 00 00 00 7d 00 00 00
0x01d0   41 fe 06 fe ff fb 7e 04-7d 00 7e 04 7d 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 55 aa
```

The new 4 GB partition is at offset 0x007D047E (8,193,150). This is *not* aligned. Apparently `diskpart` behaves differently to the Disk Manager GUI and will force the start of a partition to be cylinder-aligned,

whereas the Disk Manager GUI just took the next available block for the partition start. Note that you now have a "hole" between the end of the first partition (0x007D0080) and the start of the second partition (0x007D047E).

## Conclusion

You can use `diskpar` to align a volume on a Windows 2000 Basic Disk. The steps are:

1. Ensure the disk is a Basic Disk. Delete all volumes and revert to Basic Disk if necessary.

2. Using `diskpar`, create the desired partition size aligned at one of the following:
   - 128 blocks if a Symmetrix volume
   - The CLARiiON element size if a CLARiiON volume

3. You *can* create one more subsequent volume that is aligned *if* you are using the Disk Management GUI. Always check the MBR partition entry to ensure this.

4. You *cannot* create any more partitions with `diskpart` and expect them to be aligned.

Note that this is different for Windows 2003. Under Windows 2003, you can only create one partition with `diskpar` that is aligned. Creating more partitions using the GUI or `diskpart` creates nonaligned partitions.

# Using diskpar with Windows 2003 Basic Disks

Starting with a completely empty Symmetrix volume of 18414 cylinders, the following sections present some interesting blocks as the steps are performed. Use the `secinspect` tool from the Windows 2000/2003 Resource Kit.

| Block | Contents |
|---|---|
| Block 0 | Master Boot Record. |
| Block 63 | Default block for the first partition created. This is aligned on a track boundary where the definition of a track is 63 blocks. |
| Block 128 | Block to use for the partition because it is aligned on a 128-block boundary. If this was a CLARiiON array, you would most likely be looking at block 128. |

## *Step 1: Write the Disk Signature*

After the signature has been written, the MBR contains the following:

**Block 0 – Master Boot Record**

```
LBN 0

******
0x01b0   00 00 00 00 00 00 00 00-7a 7d 88 01 00 00 00 00
0x01c0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This shows a valid signature (0x01887D7A) and no partitions defined.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

**Block 128 – The (aligned) block address on which to create a partition**

```
LBN 128

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

## Step 2: Run diskpar and Create a Partition

This example shows how to run `diskpar` to create a 4 GB partition.

```
C:\ diskpar -s 1
Set partition can only be done on a raw drive.
You can use Disk Manager to delete all existing partitions
Are you sure drive 1 is a raw device without any partition? (Y/N) y

---- Drive 1 Geometry Information ----
Cylinders = 1174
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 9656478720 (Bytes) = 9209 (MB)

We are going to set the new disk partition.
All data on this drive will be lost. continue (Y/N)? y

Please specify starting offset (in sectors): 128
Please specify partition length (in MB) (Max = 9209): 4000

Done setting partition.
---- New Partition information ----
StatringOffset = 65536
PartitionLength = 4194304000
HiddenSectors = 128
PartitionNumber = 1
PartitionType = 7
```

You should now use Disk Manager to format this partition.

Note that, unlike Windows 2000, the Windows 2003 Disk Management GUI *does* update automatically with the new partition information if it was open when `diskpar` was run.

## Step 3: Format the 4 GB Partition

After formatting the MBR contains the following:

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-7a 7d 88 01 00 00 00 01
0x01c0   02 00 07 ed 70 fd 80 00-00 00 00 00 7d 00 80 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 80 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The offset to the partition is 0x00000080 (128), which is what you specified in `diskpar`.

The length of the partition is 0x007D0000 (8,192,000) blocks (4 GB).

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

The zeroes indicate that the block has not been touched. The newly created partition definitely does not start here.

**Block 128 – The (aligned) block address where you created the 4 GB partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   ÙRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff ff 7c 00 00 00 00 00   ....Ç.Ç.  |.....
0x0030   00 00 04 00 00 00 00 00-ff cf 07 00 00 00 00 00   ........ ¤......
0x0040   f6 00 00 00 01 00 00 00-b7 a2 2f a8 dd 2f a8 ae   ÷.......Àó/¿¦/¿«
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ãð+.|¹©+.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   ÄÏÞ?.©..Ä+3¦ã...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?ÞS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.Âã@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .ÂÐÇÔ?¸Ôå-+Ý.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   À+f¸ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.ü¹U¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..Þ¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï¶-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?Ù-f3Êf.À.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f¸±¦-è-fïðf-Û?¸6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.åÍè?$.èÞ+õ..¦©
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.Þ..á¹.Þ..¹Ù¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï-¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   Ù=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

This is the NTFS boot record for the new volume. Note the value at offset 0x28. This is the size of the partition in blocks. The value is 0x007CFFFF (8,191,999). This is one less than the actual size you specified, and it ensures that the backup copy of the NTFS boot record (the last block in this partition) is untouched.

If you look at block 128 with DskProbe2, you see the following:

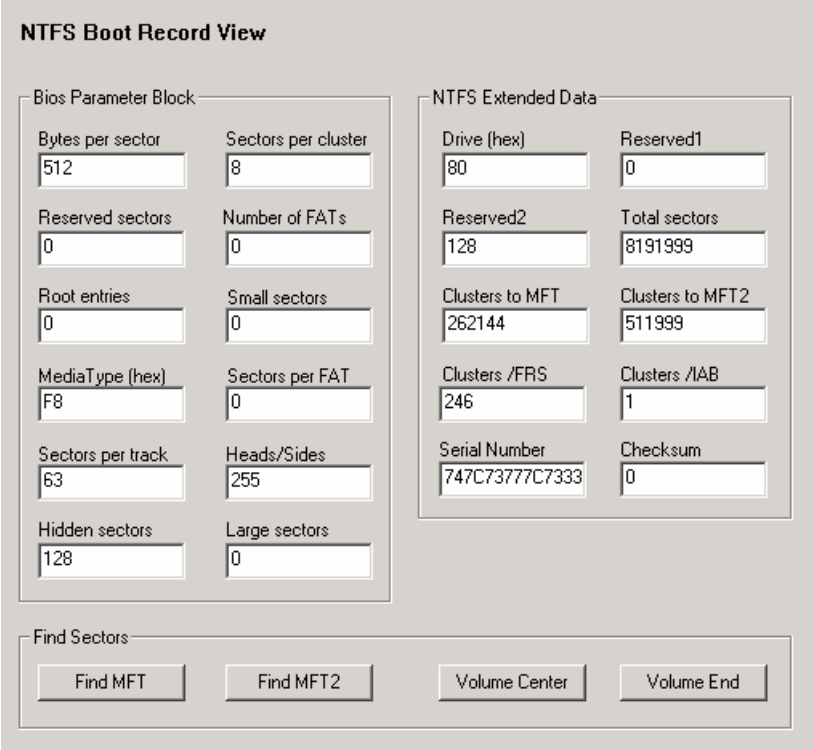



**Figure 7. Viewing the NTFS Boot Record**

This looks like a good NTFS boot record. The partition length checks out, but as a further check, look at the backup NTFS boot record located at the last block in the partition:

Offset to this partition (128) + Size of partition (8,192,000) – 1 = 8,192,127

```
LBN 8192127

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   ÙRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff ff 7c 00 00 00 00 00   ....Ç.Ç.  |.....
0x0030   00 00 04 00 00 00 00 00-ff cf 07 00 00 00 00 00   ........ ¤......
0x0040   f6 00 00 00 01 00 00 00-b7 a2 2f a8 dd 2f a8 ae   ÷.......Àó/¿¦/¿«
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ãð+.|¹©+.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   ÄÏÞ?.©..Ä+3¦ã...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?ÞS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.Âã@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .ÂÐÇÔ?¸Ôâ-+Ý.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   À+f¸ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.ü¹U¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..Þ¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï¶-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?Ù-f3Ðf.À.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f¸±¦-è-fïðf-Û?¸6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.åÍè?$.èÞ+õ..¦©
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.Þ..á¹.Þ..¹Ù¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï-¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   Ù=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

It is the same as the NTFS boot record at block 128. So this is the last block of the partition.

## Step 4: Create Another 4 GB Partition and Check the Alignment

You must use the LDM GUI to do this (or the `diskpart` utility). Do not use `diskpar` since this will destroy the first partition you just created.

Create another 4 GB partition from the LDM GUI.

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-7a 7d 88 01 00 00 00 01
0x01c0   02 00 07 ed 70 fd 80 00-00 00 00 00 7d 00 00 00
0x01d0   41 fe 06 fe ff fb 7e 04-7d 00 7e 04 7d 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The new 4 GB partition is at offset 0x007D047E (8,193,150). This second partition is *not* aligned but is divisible by 63.

Note that this is different than Windows 2000—remember that the Windows 2000 Logical Disk GUI created a second partition that *was* aligned.

The length of this new partition coincidentally is the same as the offset.

Windows 2003 has adjusted the starting offset to this second partition to be (63 block) cylinder-aligned, whereas Windows 2000 just set the offset to the next available block.

With Windows 2003, there is a gap between the two partitions 0x007D0080 to 0x007D047D.

But what if you create another partition?

Create another 1 GB partition from the LDM GUI.

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-7a 7d 88 01 00 00 00 01
0x01c0   02 00 07 ed 70 fd 80 00-00 00 00 00 7d 00 00 00
0x01d0   41 fe 06 fe ff fb 7e 04-7d 00 7e 04 7d 00 00 00
0x01e0   c1 fc 06 fe ff ff fc 08-fa 00 80 60 1f 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This third partition is *not* aligned either. The offset is 0x00FA08FC (16,386,300) and is not divisible by 128. It *is* divisible by 63 though as would be expected.

## Step 5: Create the Second 4 GB Partition with diskpart and Check the Alignment

If you delete the last partition you just created (or run `diskpar` again), you can create a second 4 GB partition using `diskpart` instead of the Disk Manager GUI.

```
C:\ diskpart

Microsoft DiskPart version 1.0
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: VM-SRV1

DISKPART> select disk 1
```

```
Disk 1 is now the selected disk.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           4000 MB    64 KB

DISKPART> create partition primary size=4000

DiskPart succeeded in creating the specified partition.

DISKPART> list  partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           4000 MB    64 KB
* Partition 2    Primary           4001 MB  4001 MB

DISKPART>
```

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-a9 a0 94 a5 00 00 00 01
0x01c0   02 00 07 ed 70 fd 80 00-00 00 00 00 7d 00 00 00
0x01d0   41 fe 06 fe ff fb 7e 04-7d 00 7e 04 7d 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The new 4 GB partition is at offset 0x007D047E (8,193,150). This is *not* aligned either.

Windows 2003 always aligns partitions it creates to be (63 block) cylinder-aligned.

## *Conclusion*

You can use `diskpar` to align a volume on a Windows 2003 Basic Disk. The steps are:

1. Ensure the disk is a Basic Disk. Delete all volumes and revert to Basic Disk if necessary.

2. Using `diskpar`, create the desired partition size aligned at one of the following:
   - 128 blocks if a Symmetrix volume
   - The CLARiiON element size if a CLARiiON volume

3. You *cannot* create any more subsequent volumes that are aligned.

Note that this is different for Windows 2000. Under Windows 2000, you can create one extra partition that is aligned by using the Disk Management GUI (but not `diskpart`).

# Using diskpart with Windows 2003 SP1 Basic Disks

Starting with a completely empty Symmetrix volume of 18414 cylinders, the following sections present some interesting blocks as the steps are performed. Use the `secinspect` tool from the Windows 2000/2003 Resource Kit.

| Block | Contents |
|---|---|
| Block 0 | Master Boot Record. |
| Block 63 | Default block for the first partition created. This is aligned on a track boundary where the definition of a track is 63 blocks. |
| Block 128 | Block to use for the partition because it is aligned on a 128-block boundary. If this was a CLARiiON array, you would most likely be looking at block 128. |

## *Step 1: Write the Disk Signature*

After the signature has been written, the MBR contains the following:

**Block 0 – Master Boot Record**

```
LBN 0

******
0x01b0   00 00 00 00 00 00 00 00-f0 08 e6 2c 00 00 00 00
0x01c0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This shows a valid signature (0x2CE608F0) and no partitions defined.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

**Block 128 – The (aligned) block address on which to create a partition**

```
LBN 128

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

## Step 2: Run diskpart and Create a Partition

This example shows how to run `diskpart` to create a 4 GB partition on disk 1.

```
C:\ diskpart

Microsoft DiskPart version 5.2.3790.1830
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: W3K-LAB

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> create partition primary align=64

DiskPart succeeded in creating the specified partition.

DISKPART> exit

Leaving DiskPart...

C:\
```

Note that the value that is used for alignment is expressed in Mbytes and not blocks as in `diskpar`. You should now use Disk Manager to format this partition

Note that, unlike Windows 2000, the Windows 2003 Disk Management GUI *does* update automatically with the new partition information if it was open when `diskpart` was run.

## Step 3: Format the 4 GB Partition

After formatting the MBR contains the following:

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-f0 08 e6 2c 00 00 00 02
0x01c0   03 00 07 fe bf 09 80 00-00 00 fe 03 7d 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The offset to the partition is 0x00000080 (128), which is what you specified in `diskpart`.

The length of the partition is 0x007D03FE (8,193,022) blocks (a little above 4 GB). The documentation for `diskpart` states that `diskpart` rounds the size allocation to a cylinder boundary, i.e.,

8.183,022 + 128 (8,193,150) which is 510 (cylinders) * 255 (heads) * 63 (sectors per track)

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

The zeroes indicate that the block has not been touched. The newly created partition definitely does not start here.

**Block 128 – The (aligned) block address where you created the 4 GB partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   ÙRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .Ç...
0x0020   00 00 00 00 80 00 80 00-fd 03 7d 00 00 00 00 00   ....Ç.Ç.².}.....
0x0030   00 00 04 00 00 00 00 00-3f d0 07 00 00 00 00 00   ........?ð......
0x0040   f6 00 00 00 01 00 00 00-08 86 f5 f0 8d f5 f0 44   ÷........å§-ì§-D
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3└Äð╝.|¹©└.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   ÄÏÞ─.©..Ä└3╟ã...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ►ÞS.h..hj.╦è─$.┤
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .=‼s.╣  è±f.Âã@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .ÂÐÇÔ?¸Ôå═└Ý.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   À╔f¸ßfú .├┤A╗¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ─$.═‼r.ü¹U¬u.÷┴.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.■.¶.├f`▲.fí►.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..∟.f;. ..é:.▲fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh►...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..Þ│ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ┤Bè─$.─▼ï¶=‼fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX▼Ù-f3╥f.À.↑.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f¸±■┬è╩fïðf┴Û▶¸6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   →.åÍè─$.èÞ└õ..╠©
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..═‼.é↓.î└. .Ä└f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .►. ....àο .▼fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   ├á°.Þ..á¹.Þ..¹Ù■
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ┤.ï¬¼<.t.┤.╗..═►
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   Ù▁├..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá│╔..U¬
```

This is the NTFS boot record for the new volume. Note the value at offset 0x28. This is the size of the partition in blocks. The value is 0x007D03FD (8,193,021). This is one less than the actual size you specified, and it ensures that the backup copy of the NTFS boot record (the last block in this partition) is untouched.

If you look at block 128 with DskProbe2, you see the following:

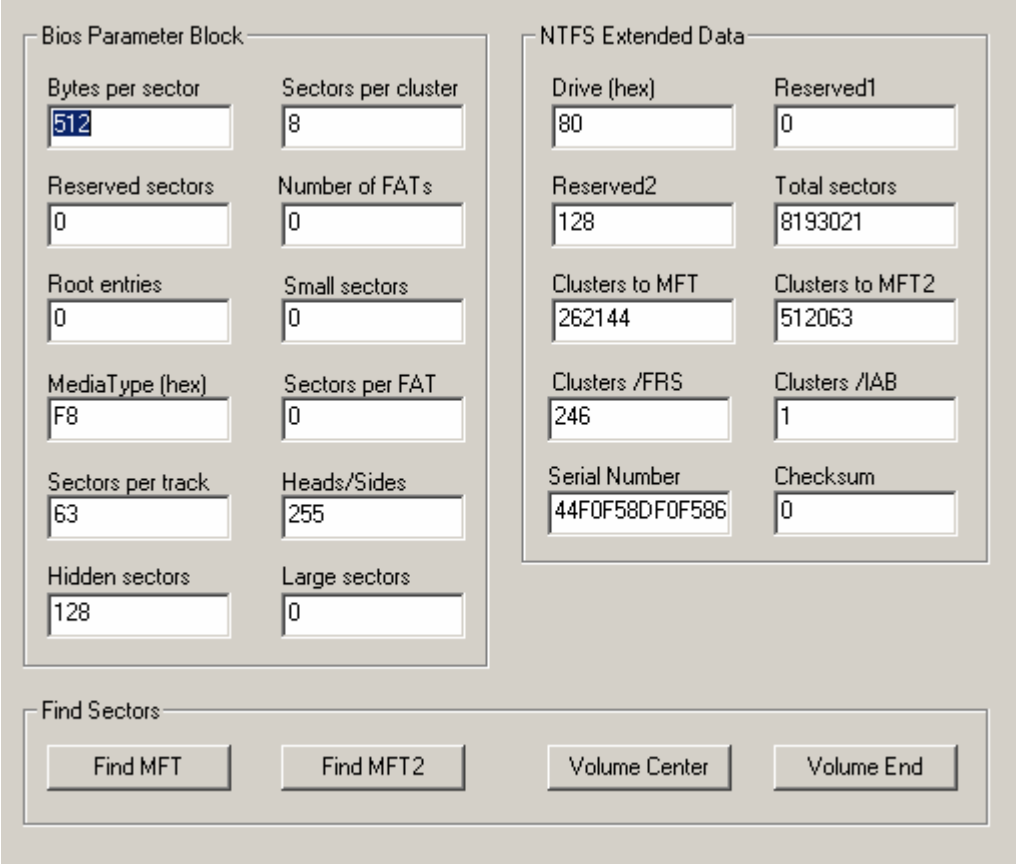



**Figure 8. Viewing the NTFS Boot Record**

This looks like a good NTFS boot record. The partition length checks out, but as a further check, look at the backup NTFS boot record located at the last block in the partition:

Offset to this partition (128) + Size of partition (8,193,022) – 1 = 8,193,149

```
LBN 8193149

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   ÙRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .Ç...
0x0020   00 00 00 00 80 00 80 00-fd 03 7d 00 00 00 00 00   ....Ç.Ç.².}.....
0x0030   00 00 04 00 00 00 00 00-3f d0 07 00 00 00 00 00   ........?ð......
0x0040   f6 00 00 00 01 00 00 00-08 86 f5 f0 8d f5 f0 44   ÷........å§-ì§-D
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3└Äð╝.|¹©└.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   ÄÏÞ─.©..Ä└3╚ã...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ►ÞS.h..hj.╦è─$.┤
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .=‼s.╣  è±f.Âã@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .ÂÐÇÔ?¸Ôå═└ý.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   À╔f¸ßfú .├┤A╗¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ─$.=‼r.ü¹U¬u.÷┴.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.■.¶.├f`▲.fí►.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..∟.f;. ..é:.▲fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh►...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..Þ│ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ┤Bè─$.─▼ï¶=‼fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX▼Ù-f3╥f.·.↑.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f¸±■┬è╦fïðf┴Û►¸6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   →.åÍè─$.èÞ└õ..╠©
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..=‼.é↓.î└. .Ä└f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .►. ....ào .▼fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   ├á°.Þ..á¹.Þ..¹Ù■
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ┤.ï¬¼<.t.┤.╖..═►
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   Ù▁├..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
```

```
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá│╔..U¬
```

It is the same as the NTFS boot record at block 128. So this is the last block of the partition.

## *Step 4: Create Another 4 GB Partition and Check the Alignment*

You must use the LDM GUI to do this (or the `diskpart` utility). Do not use `diskpar` since this will destroy the first partition you just created.

Creating another 4 GB partition using either `diskpart` or the GUI will yield the following result. Note that unfortunately you cannot use the align option with `diskpart` as this can only be used on the first partition on the disk.

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-7a 7d 88 01 00 00 00 01
0x01c0   02 00 07 ed 70 fd 80 00-00 00 fe 03 7d 00 00 00
0x01d0   41 fe 06 fe ff fb 7e 04-7d 00 7e 04 7d 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The new 4 GB partition is at offset 0x007D047E (8,193,150). This second partition is *not* aligned but is divisible by 63.

Note that this is different than Windows 2000—remember that the Windows 2000 Logical Disk GUI created a second partition that *was* aligned.

The length of this new partition coincidentally is the same as the offset.

Windows 2003 has adjusted the starting offset to this second partition to be (63 block) cylinder-aligned, whereas Windows 2000 just set the offset to the next available block.

But what if you create another partition?

Create another 1 GB partition from the LDM GUI.

```
LBN 0

0x01b0   00 00 00 00 00 00 00 00-7a 7d 88 01 00 00 00 01
0x01c0   02 00 07 ed 70 fd 80 00-00 00 00 00 7d 00 00 00
0x01d0   41 fe 06 fe ff fb 7e 04-7d 00 7e 04 7d 00 00 00
0x01e0   c1 fc 06 fe ff ff fc 08-fa 00 80 60 1f 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This third partition is *not* aligned either. The offset is 0x00FA08FC (16,386,300) and is not divisible by 128. It *is* divisible by 63 though as would be expected.

Windows 2003 always aligns partitions it creates to be (63 block) cylinder-aligned.

## *Conclusion*

You can use `diskpart` to align the first volume on a Windows 2003 Basic Disk. The steps are:

1. Ensure the disk is a Basic Disk. Delete all volumes and revert to Basic Disk if necessary.

2. Using `diskpart`, create the desired partition size aligned at one of the following:
   - 128 blocks if a Symmetrix volume
   - The CLARiiON element size if a CLARiiON volume
3. You *cannot* create any more subsequent volumes that are aligned.

Note that this is different for Windows 2000. Under Windows 2000, you can create one extra partition that is aligned by using the Disk Management GUI (but not `diskpart`).

# Using diskpar with Windows 2000 Dynamic Disks

Note the following:

- Use `dmdiag` version 5.1 (from Windows 2003 Support Tools) because it gives a more readable output format.
- Use the `secinspect` tool from the Windows 2000/2003 Resource Kit to look at blocks.

Starting with a completely empty Symmetrix volume of 18414 cylinders, the following sections present some interesting blocks as the steps are performed. You must run `diskpar` on a Basic Disk because `diskpar` creates a partition and not a Dynamic Volume. The steps in this section create a small (filler) partition that is aligned. When you convert the disk to a dynamic disk, the filler partition will be converted to a simple Dynamic Volume and subsequent dynamic volumes should also be aligned.

| Block | Contents |
|---|---|
| Block 0 | Master Boot Record. |
| Block 63 | Default block for the first partition created. This is aligned on a track boundary where the definition of a track is 63 blocks. |
| Block 128 | Block to use for the partition because it is aligned on a 128-block boundary. If this was a CLARiiON array, you would most likely be looking at block 128. |

## *Step 1: Write the Disk Signature*

After the signature has been written, the MBR contains the following:

**Block 0 – Master Boot Record**

```
LBN 0

******
0x01b0                          bc cf f5 96 00 00 00 00
0x01c0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This shows a valid signature (0x96F5CFBC) and no partitions defined.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

**Block 128 – The (aligned) block address on which to create the filler partition**

```
LBN 128

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

## Step 2: Run diskpar and Create an 8 MB Partition

Using the standard Windows 2000 tools, the minimum partition size that you can create is 8 MB. (Interestingly, the LDM GUI states that the minimum partition size is 7 MB, but this figure is ignored and the minimum actually created is 8 MB.) This minimum partition size might be different for different disk sizes. However, for every disk tested, the LDM GUI forced a minimum of 8 MB.

Microsoft has its reasons for restricting the creation of partitions less than 8 MB—possibly due to performance considerations. Early versions of Windows 2000 *did* ignore small partitions because the way the disk geometry was calculated led to a value of zero cylinders for the disk. This was fixed in Service Pack 2 (KB258281). So the 8 MB minimum partition size could be due to a performance or accessibility issue.

Given that we are not going to use this partition, one could argue that we could create a 1 MB partition. However, we take the cautious approach. A partition size of 8 MB is insignificant compared to the size of disks today, so we don't really save any space. Why then take a chance over such a minor amount of disk? We don't know the real reason Microsoft has enforced this restriction; only Microsoft has the authority and knowledge to state that it is safe to create a 1 MB partition (and they have not because they force the minimum partition size to be 8 MB).

```
C:\> diskpar -s 1
Set partition can only be done on a raw drive.
You can use Disk Manager to delete all existing partitions
Are you sure drive 1 is a raw device without any partition? (Y/N) y

---- Drive 1 Geometry Information ----
Cylinders = 1100
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 9047808000 (Bytes) = 8628 (MB)

We are going to set the new disk partition.
All data on this drive will be lost. continue (Y/N)? y

Please specify starting offset (in sectors): 128
Please specify partition length (in MB) (Max = 69044): 8

Done setting partition.
---- New Partition information ----
StatringOffset = 65536
PartitionLength = 8388608
HiddenSectors = 128
PartitionNumber = 1
PartitionType = 7

You should now use Disk Manager to format this partition
```

Note that, unlike Windows 2003, the Windows 2000 Disk Management GUI does not update automatically with the new partition information. If the Disk Management GUI was open when `diskpar` was run, you must restart the GUI.

## *Step 3: Format the 8 MB Partition*

After formatting, the MBR contains the following:

**Block 0 – The Master Boot Record**

```
LBN 0

******
0x01b0   00 00 00 00 00 00 00 00-bc cf f5 96 00 00 00 01
0x01c0   03 00 07 07 06 01 80 00-00 00 00 40 00 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 80 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 80 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The offset to the partition is 0x00000080 (128), which is what you specified in `diskpar`.
The length of the partition is 0x00004000 (16,384) blocks (8 MB) and is also a multiple of 128.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

The zeroes indicate that the block has not been touched. The newly created partition definitely does not start here.

**Block 128 – The (aligned) block address where you created the partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff 3f 00 00 00 00 00 00   ....Ç.Ç. ?......
0x0030   20 00 00 00 00 00 00 00-ff 1f 00 00 00 00 00 00    ....... ?......
0x0040   02 00 00 00 08 00 00 00-e4 87 99 74 af 99 74 20   ........SçÖt»Öt
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

This is the NTFS boot record for the new volume. Note the value at offset 0x28. This is the size of the partition in blocks. The value is 0x00003FFF (16,383). This is one less than the actual size you specified, and it ensures that the backup copy of the NTFS boot record (the last block in the partition) is untouched.

If you look at block 128 with DskProbe2, you see the following:

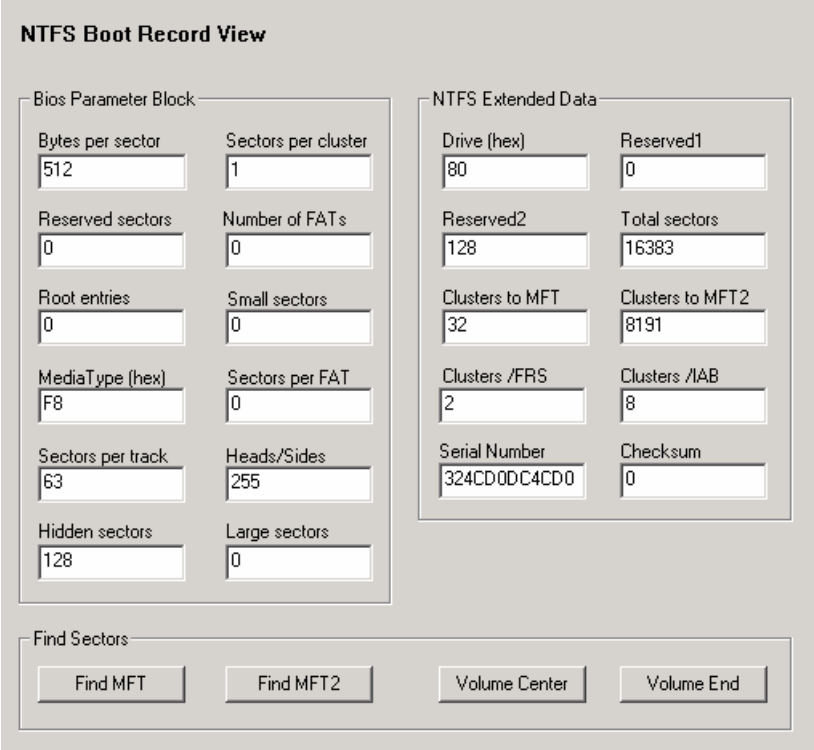



**Figure 9. Viewing the NTFS Boot Record**

It looks like a good NTFS boot record. The partition length checks out, but as a further check, look at the backup NTFS boot record located at the last block in the partition:

Offset to this partition (128) + Size of partition (16,384) – 1 = 16,511

```
LBN 16511

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff 3f 00 00 00 00 00 00   ....Ç.Ç. ?......
0x0030   20 00 00 00 00 00 00 00-ff 1f 00 00 00 00 00 00    ....... ?......
0x0040   02 00 00 00 08 00 00 00-e4 87 99 74 af 99 74 20   ........SçÖt»Öt
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
```

```
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

It is the same as the NTFS boot record at block 128. So this is the last block of this filler partition.

## *Step 4: Convert the Disk to Dynamic*

**Block 0 – The Master Boot Record**

```
LBN 0

******
0x01b0   00 00 00 00 00 2c 44 63-bc cf f5 96 00 00 00 01   .....,Dc+-)û....
0x01c0   03 00 42 07 06 01 80 00-00 00 00 40 00 00 00 06   ..B...@....@....
0x01d0   07 01 42 fe ff ff 80 40-00 00 cc 64 0d 01 00 00   ..B¦  @@...e....
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa   ..............U¬
```

When Windows 2000 converts a Basic Disk to Dynamic, any existing partitions have their address and size information unchanged in the MBR. If there are less than four partitions on the disk at the time of conversion, Windows 2000 converts the free space remaining on the disk into a new partition and places it as a new partition into the MBR. In the MBR shown previously, the original partition ended on block 0x0000407F (calculated by adding the offset [128] to the length [16384] and subtracting one).

The free space therefore starts on block 0x00004080 and the length of this free space is 0x010D64CC blocks.

Note that Windows 2000 changes the system ID in both partitions to 42.

Disks with existing partitions that are converted to Dynamic Disks are called Hard-Linked Dynamic Disks. If a Basic Disk with no existing partitions is converted, then Windows creates one partition for the entire disk. This type of Dynamic Disk is called a Pure Dynamic Disk.

This is different than Windows 2003. Converting a Basic Disk with existing partitions under Windows 2003 converts the Basic Disk into a Pure Dynamic Disk and partition information is removed from the MBR.

If you run `dmdiag`, you can see the following:

```
---------- Dynamic Disk Information -----------
 DiskGroup: Dell6Dg0
  Group-ID: 7598a143-9c46-44a4-8c15-e9f9d22b7749

   Sub Disk  Rel Sec   Tot Sec  Tot Size  Plex        Vol Type  Col/Ord  DevName      State
   ========  =======   =======  ========  ====        ========  =======  =========    ======
   Disk1-01  128       16384    17677440  Volume1-01  Simple    1/1      Harddisk1    ONLINE
   LDM-DATA  17675392  2048
```

`dmdiag` reports that:

- The 8 MB partition (Disk1-01) is *still* aligned at 128 blocks.
- The 8 MB partition size is also a multiple of 128, enabling the next volume to be aligned.
- The total size of the disk is reported as 17,677,440 blocks.
- The LDM-DATA (metadata) is also aligned and is exactly 1 MB long.

Further down, `dmdiag -v` reports the following partition table information:

```
---------- Partition Table Info Disk 17 ----------


            1100 Cylinders
             255 Tracks/Cylinder
              63 Sectors/Track
             512 Bytes/Sector
              12 MediaType
      17,671,500 Sectors (total)
```

```
    9,047,808,000 Bytes (total)
        8,835,750 KB
            8,629 MB
              8.4 GB

                0 StartingOffset
    9,050,849,280 PartitionLength
                0 HiddenSectors
                0 PartitionNumber
                0 PartitionType
                0 BootIndicator
                0 RecognizedPartition
                0 RewritePartition

              MBR PartitionStyle
                4 PartitionCount
         a16c4aaa Signature

  Starting     Partition  Hidden       Total Partition  Partition    Boot     Recognized  Rewrite
Offset (bytes)    Length Sectors     Sectors   Number   Type (HEX)  Indicator  Partition Partition

    65,536      8,388,608    128      16,384      0       0x42          0          1          0
 8,454,144  9,039,353,856 16,512  17,654,988      1       0x42          0          1          0
         0              0      0           0      2       0x00          0          0          0
         0              0      0           0      2       0x00          0          0          0

    9,047,808,000 Bytes (17671500 sectors) Geometric size
    9,050,849,280 Bytes (17677440 sectors) True size (measured)
    9,050,849,280 Bytes (17677440 sectors) Reported size (Partition0)
                0 Bytes (       0 sectors) missing/wasted
```

Note the apparent inconsistencies:

- The Master Boot Record reports the total partition sizes as: 17,671,372 Blocks (0x00004000 + 0x010D64CC)
- `dmdiag` reports the total size of the disk as: 17,677,440 Blocks
- `dmdiag` then reports the total size of the disk as: 17,671,500 Sectors
- `dmdiag` then reports the geometric size of the disk as: 17,671,500 Sectors
- `dmdiag` then reports the true (measured) size of the disk as: 17,677,440 Sectors
- The Symmetrix volume is: 18414 Cylinders
- `dmdiag` reports cylinder count as: 1100 Cylinders

The difference in these reported numbers is a consequence of the Symmetrix volume geometry being different than the original IBM PC-BIOS (from which Windows disk layouts have descended).

Symmetrix tracks are defined tracks being 64 blocks.

Symmetrix cylinders are defined as being 15 heads.

Windows disk tracks are defined as being 63 blocks.

Windows disk cylinders are defined as being 255 heads.

When a Symmetrix volume is created, it is always created in multiples of cylinders (15 Heads * 64 Tracks). For example, if you need a volume of 9 GB, you actually get allocated 18,414 cylinders.

The actual number of blocks in this volume therefore is (18414 * 15 * 64) = 17,677,440 blocks.

So `dmdiag` gets the correct figure for the true (measured) size of the volume.

When Windows needs to calculate the number of cylinders, it is calculated as follows (Microsoft KB258281):

```
Number of Cylinders = (Reported Size in Bytes)/(Sectors per Track)/(Tracks per Cyl)/(Sector Size)
                    =  9,050,849,280 / 63 / 255 / 512
                    =  1100.369747899159663865546218487 4...
```

Taking the integral part of this result, you have 1100 cylinders (which is what `dmdiag` reported).

If you work back from this value of 1100 cylinders to calculate the number of blocks, you get the following:

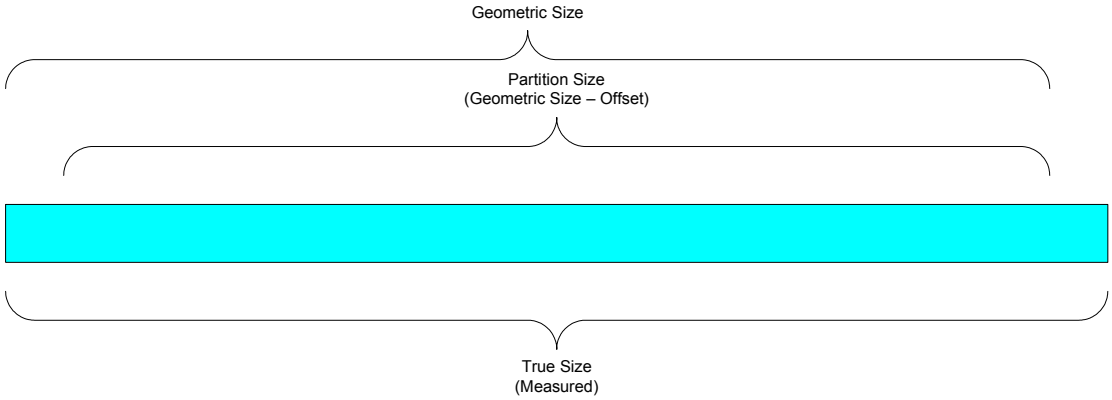
1100 Cylinders * 255 Tracks * 63 Sectors = (17,671,500 Blocks).

This is the same as the geometric size reported earlier.

If you add 128 (the offset to the first partition) to the total of the partition sizes in the MBR (17,671,372), you get 17,671,500 blocks, which is what Windows calculates as the geometric size.

The LDM-DATA (metadata) is at offset 17,675,392 (also aligned) and ends at 17,675,392 + 2047 = 17,677,439, which is the last block address of the true (measured) block of the volume.

In summary:




**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

It is still zeroes, as expected.

**Block 128 – The (aligned) block address where you created the filler partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff 3f 00 00 00 00 00 00   ....Ç.Ç. ?......
0x0030   20 00 00 00 00 00 00 00-ff 1f 00 00 00 00 00 00    ....... ?......
0x0040   02 00 00 00 08 00 00 00-e4 87 99 74 af 99 74 20   ........SçÖt»Öt
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
```

```
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

It is still there. *The dynamic filler volume is still aligned* just as `dmdiag` indicated.

**Block 16511 – The backup NTFS boot record**

```
LBN 16511

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff 3f 00 00 00 00 00 00   ....Ç.Ç. ¦......
0x0030   aa 2a 00 00 00 00 00 00-ff 1f 00 00 00 00 00 00   ¬*...... ?......
0x0040   02 00 00 00 08 00 00 00-32 71 86 20 88 86 20 54   ........2qå êå T
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.ùvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

The backup NTFS boot block is correct as well. So the original partition is still there, but now is a Dynamic Simple Volume.

## *Step 5: Create More Volumes and Check Their Alignments*

Now create the volume(s) that you will actually use. The first volume is just a filler volume that you needed to align subsequent volumes.

Create a 1 GB Simple Volume and format it.

`dmdiag` now reports:

```
---------- Dynamic Disk Information -----------
 DiskGroup: Dell9Dg0
  Group-ID: 7598a143-9c46-44a4-8c15-e9f9d22b7749

   Sub Disk  Rel Sec  Tot Sec  Tot Size  Plex        Vol Type  Col/Ord  DevName    State
   ========  =======  =======  ========  ====        ========  =======  =========  ======
   Disk1-01  128       16384    17677440  Volume1-01  Simple    1/1  Harddisk1   ONLINE
   Disk1-02  16512     2048000  17677440  Volume2-01  Simple    1/1  Harddisk1   ONLINE
   LDM-DATA  17675392  2048
```

The second volume is aligned at block 16,512, which is also aligned.

**Block 0 – The Master Boot Record**

```
0x01b0   00 00 00 00 00 2c 44 63-bc cf f5 96 00 00 00 01
0x01c0   03 00 42 07 06 01 80 00-00 00 00 40 00 00 00 06
0x01d0   07 01 42 fe ff ff 80 40-00 00 cc 64 0d 01 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

As expected, the MBR is unchanged. Once the disk is converted to a Dynamic Disk, the MBR information almost never changes. Creating a mirror for a volume that is on a Hard-Linked Dynamic disk to another Hard-Linked Dynamic Disk *will* create a new partition in the MBR on the target disk for the second piece of the volume (Microsoft KB242436).

From `dmdiag` earlier, you can see that the start of the new volume is 16,512. This is an aligned address. The size of this volume is 2,048,000, which is a multiple of 128.

**Block 16512 –The (aligned) block address where Windows created the 1 GB Simple Dynamic Volume**

```
LBN 16512

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 02 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 1d 00 00 00   .....°..?. .?...
0x0020   00 00 00 00 80 00 80 00-ff 3f 1f 00 00 00 00 00   ....Ç.Ç. ??.....
0x0030   10 00 00 00 00 00 00 00-ff cf 07 00 00 00 00 00   ?....... -......
0x0040   01 00 00 00 04 00 00 00-4d 54 7d 74 78 7d 74 9e   ........MT}tx}tP
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

Note the contents at offset 0x28. The value is 0x0001F3FF which is 2,047,999. If you add 1 to this, you get 2,048,000 blocks, which is 1 GB. This is the Dynamic Volume you created, and it is aligned.

As more volumes are created on this disk, *they are all aligned correctly*. This is because Windows considers a Megabyte to be exactly 2048 blocks (nicely divisible by 128).

## Conclusion

You can use `diskpar` to align volumes on a Windows 2000 Dynamic Disk. The steps are:

1. Ensure the disk is a Basic Disk. Delete all volumes and revert to Basic Disk if necessary.
2. Using `diskpar`, create an 8 MB partition aligned at one of the following:
   - 128 blocks if a Symmetrix volume
   - The CLARiiON element size if a CLARiiON volume
3. Convert the Basic Disk to a Dynamic Disk.
4. Subsequent volumes are aligned, but always use `dmdiag` to ensure this.

Use `dmdiag` Version 5.1 from Window 2003—it gives a more readable output.

`dmdiag` can apparently give conflicting results, but they can be explained.

# Using diskpar with Windows 2003 Dynamic Disks

Note the following:

- Use `dmdiag` version 5.1 (from Windows 2003 Support Tools) because it gives a more readable output format.
- Use the `secinspect` tool from the Windows 2000/2003 Resource Kit to look at blocks.

Starting with a completely empty Symmetrix volume of 958 cylinders, the following sections present some interesting blocks as the steps are performed. You must run `diskpar` on a Basic Disk because `diskpar` creates a partition and not a Dynamic Volume; then, follow the steps in this section to create a small (filler) partition that is aligned. When you convert the disk to a dynamic disk, this filler partition will be converted to a simple Dynamic Volume and subsequent dynamic volumes should also be aligned.

| Block | Contents |
|---|---|
| Block 0 | Master Boot Record. |
| Block 63 | Default block for the first partition created. This is aligned on a track boundary where the definition of a track is 63 blocks. |
| Block 128 | Block to use for the partition because it is aligned on a 128-block boundary. If this was a CLARiiON array, you would most likely be looking at block 128. |

## *Step 1: Write the Disk Signature*

After the signature has been written, the MBR contains the following:

**Block 0 – Master Boot Record**

```
LBN 0

******
0x01b0                              aa 4a 6c a1 00 00 00 00
0x01c0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This shows a valid signature (0xA16C4AAA) and no partitions defined.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

**Block 128 – The (aligned) block address on which to create the filler partition**

```
LBN 64

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

## Step 2: Run diskpar and Create a 16 MB or 24 MB Partition

Using the standard Windows 2003 tools, some disks require a minimum partition size of 16 MB. This minimum partition size can be different for different disk sizes. On one system tested, the LDM GUI forced a minimum of 24 MB.

Microsoft has its reasons for restricting the creation of small partitions and it is possibly due to performance considerations. Early versions of Windows 2000 *did* ignore small partitions because the way the disk geometry was calculated led to a value of zero cylinders for the disk. Given that we are not going to use this partition, one could argue that we could create a 1 MB partition. However, I take the cautious approach. 16 (or 24) MB is insignificant compared to the size of disks today, so we don't really save any space. Why then take a chance over such a minor amount of disk? We don't know the real reason Microsoft has enforced this restriction, only Microsoft has the authority and knowledge to state that it is safe to create a 1 MB partition (and they have not because they force the minimum partition size to be larger).

It is recommended that you determine the minimum partition for a particular disk and use that value. The following example uses a minimum partition size of 16 MB.

```
C:\ diskpar -s 1
Set partition can only be done on a raw drive.
You can use Disk Manager to delete all existing partitions
Are you sure drive 1 is a raw device without any partition? (Y/N) y

---- Drive 1 Geometry Information ----
Cylinders = 57
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 468840960 (Bytes) = 447 (MB)

We are going to set the new disk partition.
All data on this drive will be lost. continue (Y/N)? y

Please specify starting offset (in sectors): 128
Please specify partition length (in MB) (Max = 447): 16

Done setting partition.
---- New Partition information ----
StatringOffset = 65536
PartitionLength = 16777216
HiddenSectors = 128
PartitionNumber = 1
PartitionType = 7

You should now use Disk Manager to format this partition
```

Note that unlike Windows 2000, Windows 2003 Disk Management GUI updates automatically with the new partition information and you do not have to restart the Disk Management GUI.

## Step 3: Format the 16 MB Partition

After formatting, the MBR contains the following:

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0                          ..... aa 4a 6c a1 00 01
0x01c0   03 00 07 0c 0a 02 80 00-00 00 00 80 00 00 80 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 80 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The offset to the partition is 0x00000080 (128), which is what you specified in `diskpar`.

The length of the partition is 0x00008000 (32,768) blocks (16 MB) and is also a multiple of 128.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

The zeroes indicate that the block has not been touched. The partition definitely does not start here.

**Block 128 – The (aligned) block address where you created the partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff 7f 00 00 00 00 00 00   ....Ç.Ç. ¦.......
0x0030   aa 2a 00 00 00 00 00 00-ff 3f 00 00 00 00 00 00   ¬*...... ?......
0x0040   02 00 00 00 08 00 00 00-32 71 86 20 88 86 20 54   ........2qå êå T
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

This is the NTFS boot record for the new volume. Note the value at offset 0x28. This is the size of the partition in blocks. The value is 0x00007FFF (32,767). This is one less than the actual size you specified, and it ensures that the backup copy of the NTFS boot record (the last block in the partition) is untouched.

This is expected. This is exactly what was done above with Windows 2000 and Windows 2003 Basic Disks.

## Step 4: Convert the Disk to Dynamic

After converting the disk to dynamic, the MBR is as follows:

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0                                  d2 b7 aa 4a 6c a1 00 01
0x01c0   01 00 42 fe 3f 38 3f 00-00 00 ba f8 0d 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The MBR shows something very interesting. Converting to a Dynamic Disk *appears* to have changed the offset and length of partition you created. The offset is set back to the default (0x0000003F) and the length is set to 0x000DF8BA blocks (915,642) or 468,808,704 bytes, which is close to the actual size of the disk (447 MB). The partition type is set to 0x42—a Dynamic Disk.

Under Windows 2000, converting a Basic Disk to a Dynamic Disk leaves the partition information in the MBR untouched, but Windows 2003 converts the disk to a true Dynamic Disk.

If you run `dmdiag`, you can see the following:

```
---------- Dynamic Disk Information -----------
 DiskGroup: Dell9Dg0
  Group-ID: 7598a143-9c46-44a4-8c15-e9f9d22b7749

   Sub Disk  Rel Sec   Tot Sec  Tot Size  Plex        Vol Type  Col/Ord  DevName    State
   ========  =======   =======  ========  ====        ========  =======  =========  ======
   Disk1-01  128         32768   919680    Volume1-01  Simple    1/1      Harddisk1  ONLINE
   LDM-DATA  917632      2048
```

`dmdiag` reports that:

- The 16 MB partition (Disk1-01) is *still* aligned at 128 blocks.
- The total size of the disk is reported as 919,680 blocks.
- The LDM-DATA (metadata) at offset 917,632 is also aligned (divisible by 128) and is exactly 1 MB long.

Further down, `dmdiag -v` reports the following partition table information:

```
---------- Partition Table Info Disk 2 ----------


             57 Cylinders
            255 Tracks/Cylinder
             63 Sectors/Track
            512 Bytes/Sector
             12 MediaType
        915,705 Sectors (total)
    468,840,960 Bytes (total)
        457,853 KB
            447 MB
            0.4 GB

              0 StartingOffset
    470,876,160 PartitionLength
              0 HiddenSectors
              0 PartitionNumber
              0 PartitionType
              0 BootIndicator
              0 RecognizedPartition
              0 RewritePartition

            MBR PartitionStyle
              4 PartitionCount
       a16c4aaa Signature
```

```
Starting       Partition  Hidden     Total Partition  Partition       Boot Recognized    Rewrite
Offset (bytes)    Length Sectors   Sectors    Number  Type (HEX) Indicator  Partition  Partition

  32,256      468,808,704      63     915,642       0       0x42         0          1          0
       0                0       0           0       1       0x00         0          0          0
       0                0       0           0       2       0x00         0          0          0
       0                0       0           0       3       0x00         0          0          0

     468,840,960 Bytes ( 915705 sectors) Geometric size
     470,876,160 Bytes ( 919680 sectors) True size (measured)
     470,876,160 Bytes ( 919680 sectors) Reported size (Partition0)
               0 Bytes (       0 sectors) missing/wasted
```

Note the apparent inconsistencies:

- The Master Boot Record reports the disk partition size as: 915,642 Blocks
- `dmdiag` reports the total size of the disk as: 919,680 Blocks
- `dmdiag` then reports the total size of the disk as: 915,705 Sectors
- `dmdiag` then reports the geometric size of the disk as: 915,705 Sectors
- `dmdiag` then reports the true (measured) size of the disk as: 919,680 Sectors
- The Symmetrix cylinder count is: 958 Cylinders
- `dmdiag` reports the cylinder count as: 57 Cylinders

The difference in these reported numbers is a consequence of the Symmetrix volume geometry being different than the original IBM PC-BIOS (from which Windows disk layouts have descended).

Symmetrix tracks are defined tracks being 64 blocks.

Symmetrix cylinders are defined as being 15 heads.

Windows disk tracks are defined as being 63 blocks.

Windows disk cylinders are defined as being 255 heads.

When a Symmetrix volume is created, it is always created in multiples of cylinders (15 Heads * 64 Tracks). For example, if you need a volume of 447 MB, you actually get allocated 958 cylinders. The actual number of blocks in this volume therefore is (958 * 15 * 64) = 919,680 blocks.

So `dmdiag` gets the correct figure for the true (measured) size of the volume.

When Windows needs to calculate the number of cylinders, it is calculated as follows (KB258281):

```
Number of Cylinders = (Reported Size in Bytes)/(Sectors per Track)/(Tracks per Cyl)/(Sector Size)
                    =  470,876,160 / 63 / 255 / 512
                    =  57.247432306255835667600373482726...
```

Taking the integral part of this result, you have 57 cylinders (which is what `dmdiag` reported).

If you work back from this value of 57 cylinders to calculate the number of blocks, you get the following:

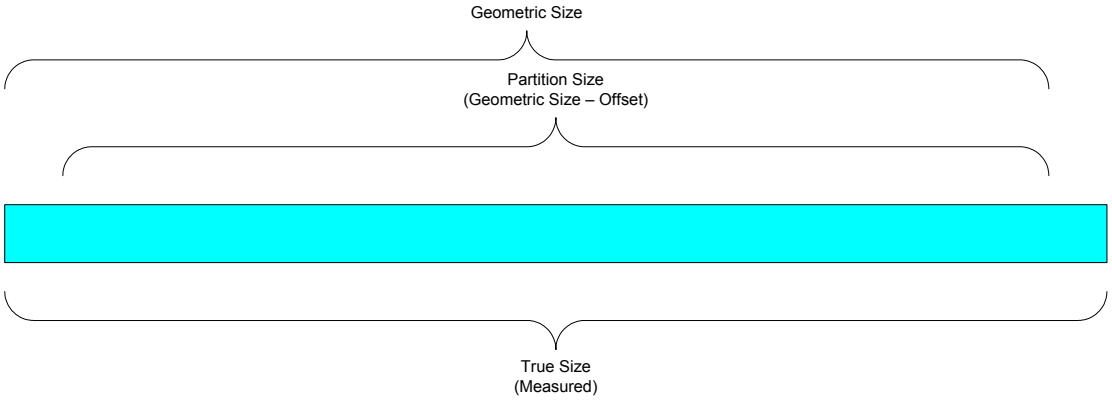
57 Cylinders * 255 Tracks * 63 Sectors = (915,705 Blocks)

This is the same as the geometric size reported earlier.

If you add 63 (the offset to the first partition) to the total of the partition sizes in the MBR (915,642), you get 915,705 blocks, which is what Windows calculates as the geometric size.

The LDM-DATA (metadata) is at offset 917,632 (aligned) and ends at 917,632 + 2047 = 917,679, which is the last block of the true (measured) block of the volume.

In summary:




**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, it is still zeroes.

**Block 128 – The (aligned) block address where you created the filler partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff 7f 00 00 00 00 00 00   ....Ç.Ç. ¦......
0x0030   aa 2a 00 00 00 00 00 00-ff 3f 00 00 00 00 00 00   ¬*...... ?......
0x0040   02 00 00 00 08 00 00 00-32 71 86 20 88 86 20 54   ........2qå êå T
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

It is still there. *The dynamic filler volume is still aligned* just as `dmdiag` indicated.

**Block 32895 – The backup NTFS boot record**

```
LBN 32895

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-ff 7f 00 00 00 00 00 00   ....Ç.Ç. ¦.......
0x0030   aa 2a 00 00 00 00 00 00-ff 3f 00 00 00 00 00 00   ¬*...... ?......
0x0040   02 00 00 00 08 00 00 00-32 71 86 20 88 86 20 54   ........2qå êå T
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

The backup NTFS boot block is correct as well. The original partition is still there but is now a Dynamic Simple Volume.

## *Step 5: Create Some More Volumes and Check Their Alignments*

You can now create the volume that you will actually use. The first volume you created was just a filler volume that you needed to align subsequent volumes.

Create a 400 MB Simple Volume and format it.

`dmdiag` now reports:

```
---------- Dynamic Disk Information -----------
 DiskGroup: Dell9Dg0
  Group-ID: 7598a143-9c46-44a4-8c15-e9f9d22b7749

   Sub Disk  Rel Sec  Tot Sec  Tot Size  Plex        Vol Type  Col/Ord  DevName     State
   ========  =======  =======  ========  ====        ========  =======  =========  ======
   Disk1-01  128      32768    919680    Volume1-01  Simple    1/1      Harddisk1  ONLINE
   Disk1-02  32896    819200   919680    Volume2-01  Simple    1/1      Harddisk1  ONLINE
   LDM-DATA  917632   2048
```

The second volume is aligned at block 32,896, which is also aligned on 128 blocks.

**Block 0 – The Master Boot Record**

```
0x01b0   00 00 00 00 00 00 00 00-aa 4a 6c a1 00 00 00 01
0x01c0   01 00 42 fe 3f 38 3f 00-00 00 ba f8 0d 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

Once the disk is converted to a Dynamic Disk, the MBR information is never changed.

From the `dmdiag` output shown earlier, you can see that the start of the new volume is 32,896. This is an aligned address. The size of this volume is 819 200, which is a multiple of 128.

**Block 32896 – The (aligned) block address where you created the 400 MB Simple Dynamic Volume**

```
LBN 32896

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 02 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 3f 00 00 00   .....°..?. .?...
0x0020   00 00 00 00 80 00 80 00-ff 7f 0c 00 00 00 00 00   ....Ç.Ç. ??.....
0x0030   55 35 05 00 00 00 00 00-ff cf 07 00 00 00 00 00   U5...... -......
0x0040   01 00 00 00 04 00 00 00-9b 4a f8 60 7a f8 60 ae   ........¢J°`z°`«
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

Note the contents at offset 0x28. This is the total number of blocks in the volume. The value is 0x000C7FFF, which is 819,199. If you add 1 to this, you get 819,200 blocks, which is 400 MB.

As you create more volumes on this disk, they are all aligned correctly.

## *Conclusion*

You can use `diskpar` to align volumes on a Windows 2003 Dynamic Disk. The steps are:

1. Ensure the disk is a Basic Disk. Delete all volumes and revert to Basic Disk if necessary.

2. Using `diskpar`, create a minimum partition aligned at one of the following:
   - 128 blocks if a Symmetrix volume
   - The CLARiiON element size if a CLARiiON volume

3. Convert the Basic Disk to a Dynamic Disk.

4. Subsequent volumes are aligned. Use `dmdiag` to ensure this.

Even though the MBR appears to be incorrect, it does accurately show the start of the first available block for the Dynamic Disk area. After all, if all volumes were deleted, including the aligned filler partition, then this figure is correct and subsequent volumes (created by the LDM GUI or `diskpart` – not `diskpar`) *can* and *will* be created there.

The MBR *is* incorrect in the length of the area (as show earlier). This is unlikely to be a problem in practice.

`dmdiag` can give conflicting results but they can be explained.

# Using diskpart with Windows 2003 SP1 Dynamic Disks

Note the following:

**IMPORTANT:** You *cannot* use `diskpart` in an attempt to create aligned dynamic volumes. The following section details why this is so.

Note:

- Use `dmdiag` version 5.1 (from Windows 2003 Support Tools) because it gives a more readable output format.
- Use the `secinspect` tool from the Windows 2000/2003 Resource Kit to look at blocks.

Starting with a completely empty Symmetrix volume of 958 cylinders, the following sections present some interesting blocks as the steps are performed. Although `diskpart` can create volumes on a dynamic disk, unfortunately the align option is not available for the "Create Volume" `diskpar`t command. You must run `diskpart` on a Basic Disk to create a small (filler) partition that is aligned. When you convert the disk to a dynamic disk, this filler partition will be converted to a simple Dynamic Volume but unlike using `diskpar` the subsequent volumes are *not* aligned. The reason is that `diskpar` will create partition sizes that are based on sizes requested but `diskpart` will create partitions based on 63 sectors/track.

| Block | Contents |
|---|---|
| Block 0 | Master Boot Record. |
| Block 63 | Default block for the first partition created. This is aligned on a track boundary where the definition of a track is 63 blocks. |
| Block 128 | Block to use for the partition because it is aligned on a 128-block boundary. If this was a CLARiiON array, you would most likely be looking at block 128. |

## *Step 1: Write the Disk Signature*

After the signature has been written, the MBR contains the following:

**Block 0 – Master Boot Record**

```
LBN 0

******
0x01b0                              aa 4a 6c a1 00 00 00 00
0x01c0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This shows a valid signature (0xA16C4AAA) and no partitions defined.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

**Block 128 – The (aligned) block address on which to create the filler partition**

```
LBN 64

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, this shows all zeroes since you have not created any partitions yet.

## Step 2: Run diskpart and Create a Small Partition

Using the standard Windows 2003 tools, some disks require different minimum partition sizes. Depending on the size of the disk, the minimum partition size can as small as 8 MB, but on large disks, it may be 24 MB or larger.

Microsoft has its reasons for restricting the creation of small partitions and it is possibly due to performance considerations. Early versions of Windows 2000 *did* ignore small partitions because the way the disk geometry was calculated led to a value of zero cylinders for the disk. Given that we are not going to use this partition, one could argue that we could create a 1 MB partition. However, we take the cautious approach. A partition size of 16 (or 24) MB is insignificant compared to the size of disks today, so we don't really save any space. Why then take a chance over such a minor amount of disk? We don't know the real reason Microsoft has enforced this restriction, only Microsoft has the authority and knowledge to state that it is safe to create a 1 MB partition (and they have not because they force the minimum partition size to be larger).

It is recommended that you determine the minimum partition for a particular disk and use that value. The following example uses a minimum partition size of 16 MB.

Using the GUI or `diskpart`, you can request a 1 MB partition and the system will create the minimum size automatically.

```
C:\ diskpart

Microsoft DiskPart version 5.2.3790.1830
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: W3K-IDH

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> create partition primary size=1 align=64

DiskPart succeeded in creating the specified partition.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary             16 MB    64 KB

DISKPART> exit

Leaving DiskPart...
```

Diskpart actually created a 16 MB partition.

Note that, unlike Windows 2000, Windows 2003 Disk Management GUI updates automatically with the new partition information and you do not have to restart the Disk Management GUI.

## Step 3: Format the 16 MB Partition

After formatting, the MBR contains the following:

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0                                  ..... aa 4a 6c a1 00 01
0x01c0   03 00 07 0c 0a 02 80 00-00 00 02 7d 00 00 80 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 80 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The offset to the partition is 0x00000080 (128), which is what you specified in `diskpart`.

The length of the partition is 0x00007D02 (32,002) blocks (a bit less than 16 MB). The important point here is that it is based on a 63 sector/track boundary and not 64.

**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

The zeroes indicate that the block has not been touched. The partition definitely does not start here.

**Block 128 – The (aligned) block address where you created the partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-01 7d 00 00 00 00 00 00   ....Ç.Ç. ¦......
0x0030   aa 2a 00 00 00 00 00 00-ff 3f 00 00 00 00 00 00   ¬*...... ?......
0x0040   02 00 00 00 08 00 00 00-32 71 86 20 88 86 20 54   ........2qå êå T
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

This is the NTFS boot record for the new volume. Note the value at offset 0x28. This is the size of the partition in blocks. The value is 0x00007D01 (32,001). This is one less than the actual size you specified, and it ensures that the backup copy of the NTFS boot record (the last block in the partition) is untouched.

This is expected. This is exactly what was done above with Windows 2000 and Windows 2003 Basic Disks.

## Step 4: Convert the Disk to Dynamic

After converting the disk to dynamic, the MBR is as follows:

**Block 0 – The Master Boot Record**

```
LBN 0

0x01b0                                  d2 b7 aa 4a 6c a1 00 01
0x01c0   01 00 42 fe 3f 38 3f 00-00 00 ba f8 0d 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The MBR shows something very interesting. Converting to a Dynamic Disk *appears* to have changed the offset and length of partition you created. The offset is set back to the default (0x0000003F) and the length is set to 0x000DF8BA blocks (915,642) or 468,808,704 bytes, which is close to the actual size of the disk (447 MB). The partition type is set to 0x42—a Dynamic Disk.

Under Windows 2000, converting a Basic Disk to a Dynamic Disk leaves the partition information in the MBR untouched, but Windows 2003 converts the disk to a true Dynamic Disk.

If you run `dmdiag`, you can see the following:

```
---------- Dynamic Disk Information -----------
 DiskGroup: Dell9Dg0
  Group-ID: 7598a143-9c46-44a4-8c15-e9f9d22b7749

   Sub Disk  Rel Sec   Tot Sec  Tot Size  Plex        Vol Type  Col/Ord  DevName    State
   ========  =======   =======  ========  ====        ========  =======  =========  ======
   Disk1-01  128         32002   919680    Volume1-01  Simple    1/1      Harddisk1  ONLINE
   LDM-DATA  917632      2048
```

`dmdiag` reports that:

- The 16 MB partition (Disk1-01) is *still* aligned at 128 blocks.
- The total size of the disk is reported as 919,680 blocks.
- The LDM-DATA (metadata) at offset 917,632 is also aligned (divisible by 128) and is exactly 1 MB long.

Further down, `dmdiag -v` reports the following partition table information:

```
---------- Partition Table Info Disk 2 ----------


             57 Cylinders
            255 Tracks/Cylinder
             63 Sectors/Track
            512 Bytes/Sector
             12 MediaType
        915,705 Sectors (total)
    468,840,960 Bytes (total)
        457,853 KB
            447 MB
            0.4 GB

              0 StartingOffset
    470,876,160 PartitionLength
              0 HiddenSectors
              0 PartitionNumber
              0 PartitionType
              0 BootIndicator
              0 RecognizedPartition
              0 RewritePartition

            MBR PartitionStyle
              4 PartitionCount
       a16c4aaa Signature
```

```
Starting       Partition  Hidden     Total Partition  Partition       Boot Recognized    Rewrite
Offset (bytes)    Length Sectors   Sectors    Number  Type (HEX) Indicator  Partition  Partition

  32,256      468,808,704      63     915,642       0       0x42         0          1          0
       0                0       0           0       1       0x00         0          0          0
       0                0       0           0       2       0x00         0          0          0
       0                0       0           0       3       0x00         0          0          0

     468,840,960 Bytes ( 915705 sectors) Geometric size
     470,876,160 Bytes ( 919680 sectors) True size (measured)
     470,876,160 Bytes ( 919680 sectors) Reported size (Partition0)
               0 Bytes (       0 sectors) missing/wasted
```

Note the apparent inconsistencies:

- The Master Boot Record reports the disk partition size as: 915,642 Blocks
- `dmdiag` reports the total size of the disk as: 919,680 Blocks
- `dmdiag` then reports the total size of the disk as: 915,705 Sectors
- `dmdiag` then reports the geometric size of the disk as: 915,705 Sectors
- `dmdiag` then reports the true (measured) size of the disk as: 919,680 Sectors
- The Symmetrix cylinder count is: 958 Cylinders
- `dmdiag` reports the cylinder count as: 57 Cylinders

The difference in these reported numbers is a consequence of the Symmetrix volume geometry being different than the original IBM PC-BIOS (from which Windows disk layouts have descended).

Symmetrix tracks are defined tracks as being 64 blocks.

Symmetrix cylinders are defined as being 15 heads.

Windows disk tracks are defined as being 63 blocks.

Windows disk cylinders are defined as being 255 heads.

When a Symmetrix volume is created, it is always created in multiples of cylinders (15 Heads * 64 Tracks). For example, if you need a volume of 447 MB, you actually get allocated 958 cylinders. The actual number of blocks in this volume therefore is (958 * 15 * 64) = 919,680 blocks.

So `dmdiag` gets the correct figure for the true (measured) size of the volume.

When Windows needs to calculate the number of cylinders, it is calculated as follows (KB258281):

```
Number of Cylinders = (Reported Size in Bytes)/(Sectors per Track)/(Tracks per Cyl)/(Sector Size)
                    =  470,876,160 / 63 / 255 / 512
                    =  57.247432306255835667600373482726...
```

Taking the integral part of this result, you have 57 cylinders (which is what `dmdiag` reported).

If you work back from this value of 57 cylinders to calculate the number of blocks, you get the following:

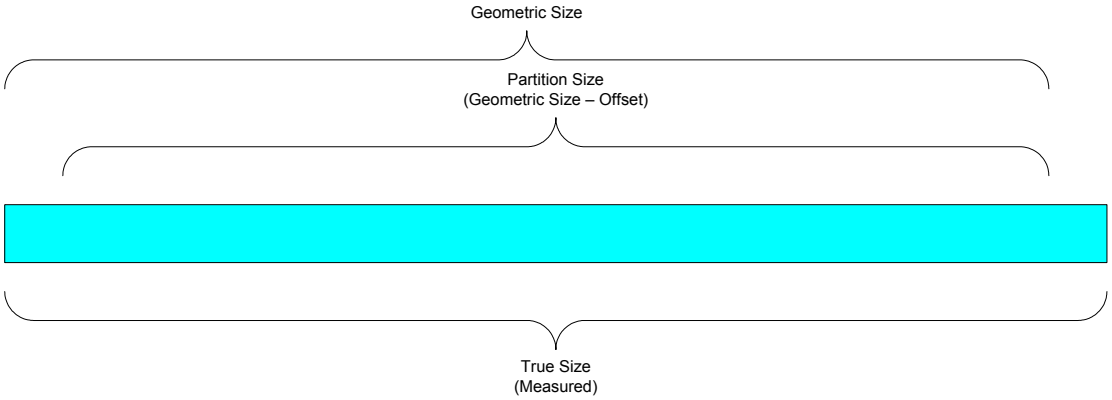
57 Cylinders * 255 Tracks * 63 Sectors = (915,705 Blocks)

This is the same as the geometric size reported earlier.

If you add 63 (the offset to the first partition) to the total of the partition sizes in the MBR (915,642), you get 915,705 blocks, which is what Windows calculates as the geometric size.

The LDM-DATA (metadata) is at offset 917,632 (aligned) and ends at 917,632 + 2047 = 917,679, which is the last block of the true (measured) block of the volume.

In summary:




**Block 63 – Default first block for an unaligned partition**

```
LBN 63

0x0000   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
******
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

As expected, it is still zeroes.

**Block 128 – The (aligned) block address where you created the filler partition**

```
LBN 128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-01 7d 00 00 00 00 00 00   ....Ç.Ç. ¦......
0x0030   aa 2a 00 00 00 00 00 00-ff 3f 00 00 00 00 00 00   ¬*...... ?......
0x0040   02 00 00 00 08 00 00 00-32 71 86 20 88 86 20 54   ........2qå êå T
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.ì+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

It is still there. *The dynamic filler volume is still aligned* just as `dmdiag` indicated.

**Block 32129 – The backup NTFS boot record**

```
LBN 32129

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 01 00 00   dRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 80 00 00 00   .....°..?. .@...
0x0020   00 00 00 00 80 00 80 00-01 7d 00 00 00 00 00 00   ....Ç.Ç. ¦......
0x0030   aa 2a 00 00 00 00 00 00-ff 3f 00 00 00 00 00 00   ¬*...... ?......
0x0040   02 00 00 00 08 00 00 00-32 71 86 20 88 86 20 54   ........2qå êå T
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ä-+.|v++.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   Ä+F?.+..Ä+3¦¦...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?FS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.¦¦@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .¦-ÇG?˜Gå-+f.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   ++f˜ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.üvU¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..F¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï(-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?d-f3-f.+.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f˜±¦-è-fï-f-O?˜6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.å+è?$.èF+S..¦+
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.F..áv.F..vd¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï=¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   d=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

The backup NTFS boot block is correct as well. The original partition is still there but is now a Dynamic Simple Volume.

## *Step 5: Create Another Volume and Check the Alignment*

You can now create the volume that you will actually use. The first volume you created was just a filler volume that you needed to attempt to align subsequent volumes.

Create a 400 MB Simple Volume and format it.

`dmdiag` now reports:

```
---------- Dynamic Disk Information -----------
 DiskGroup: Dell9Dg0
  Group-ID: 7598a143-9c46-44a4-8c15-e9f9d22b7749

   Sub Disk  Rel Sec  Tot Sec  Tot Size  Plex        Vol Type  Col/Ord  DevName     State
   ========  =======  =======  ========  ====        ========  =======  =========  ======
   Disk1-01  128      32002    919680    Volume1-01  Simple    1/1      Harddisk1  ONLINE
   Disk1-02  32130    819200   919680    Volume2-01  Simple    1/1      Harddisk1  ONLINE
   LDM-DATA  917632   2048
```

The second volume is aligned at block 32,130, which is *not* aligned on 128 blocks. It is instead aligned on a 63-track boundary.

Using `diskpart` to create a filler does *not* allow the creation of subsequent volumes to be aligned. This is because, unlike `diskpar`, `diskpart` creates partitions that are aligned on 63-track boundaries.

## *Conclusion*

You cannot use `diskpart` to align volumes on a Windows 2003 SP1 Dynamic Disk.

# Other Alignment Methods

Partitions can be aligned without using diskpar. If an Extended Partition is created, and then a logical drive created within this partition, then due to the nature of the native Windows disk layout, it turns out that this logical drive is aligned at 128 blocks.

## *Step 1: Create an Extended Partition and Logical Drive*

After creation of a 4 GB logical drive, we have the following blocks:

**Block 0 – The Master Boot Record**

```
LBN 0

******
0x01b0    00 00 00 00 00 00 00 00-aa 4a 6c a1 00 00 00 01
0x01c0    01 01 0f fe ff ff c1 3e-00 00 55 8a 1f 01 00 00
0x01d0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

When Windows creates an extended partition, it is created at a cylinder boundary. Remembering that Windows considers that cylinders are 255 heads or 63 sectors, this means that one cylinder is 16,065 blocks (255 * 63). The offset to the extended partition from the MBR is 0x00003EC1, which is 16,065 decimal.

If we look at the first block of the extended partition:

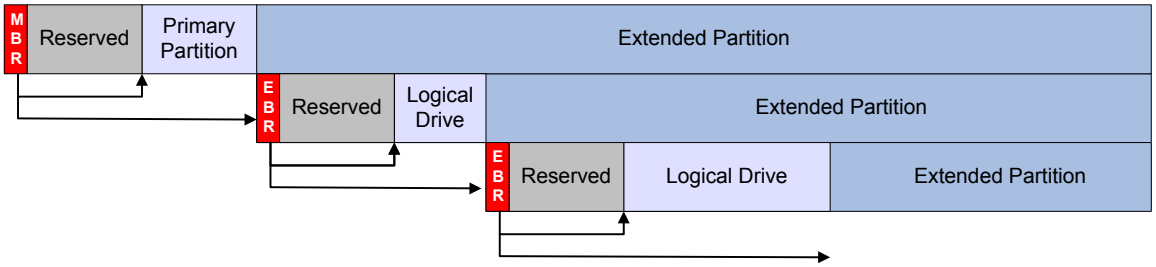
**Block 16065 – The Extended Partition Boot Record (EBR)**

```
LBN 16065

******
0x01b0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 01
0x01c0    01 01 07 fe 7f fe 3f 00-00 00 3f 04 7d 00 00 00
0x01d0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0    00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

The extended partition starts with a block that is exactly the same format as the MBR for the disk.

Essentially, extended partitions can be thought of as containing a logical drive and (possibly) another extended partition. You never have more than one logical drive per extended partition, you just have another extended partition within the first. This can be represented as:




Note that the offset to the logical drive is again 63. This offset is with respect to the start of the extended partition. This means that the logical drive starts at 16065 + 63 blocks. This value is 16,128, which just happens to be divisible by 128—*an aligned logical drive*. This can be understood by remembering that one cylinder is 255 tracks of 63 sectors per track. The offset is also a track, so we have 256 tracks of 63 sectors per track as an offset to the logical drive—clearly divisible by 128.

**Block 16128 – The block where our logical drive resides**

```
LBN 16128

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   ÙRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 3f 00 00 00   .....°..?. .?...
0x0020   00 00 00 00 80 00 80 00-3e 04 7d 00 00 00 00 00   ....Ç.Ç.>.}.....
0x0030   00 00 04 00 00 00 00 00-43 d0 07 00 00 00 00 00   ........Cð......
0x0040   f6 00 00 00 01 00 00 00-4c d0 3e a4 f0 3e a4 18   ÷.......Lð>ñ->ñ?
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Ãð+.|¹©+.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   ÄÏÞ?.©..Ä+3¦ã...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?ÞS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.Âã@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .ÂÐÇÔ?¸Ôå-+Ý.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   À+f¸ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.ü¹U¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..Þ¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï¶-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?Ù-f3Ñf.À.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f¸±¦-è-fïðf-Û?¸6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.åÍè?$.èÞ+õ..¦©
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.Þ..á¹.Þ..¹Ù¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï-¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   Ù=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

It is the NTFS boot record for the logical drive requested.

## *Step 2: Create Another Logical Drive and Check the Alignment*

After creating another 4 GB logical drive, we have the following:

**Block 0 – The Master Boot Record**

```
LBN 0

******
0x01b0   00 00 00 00 00 00 00 00-aa 4a 6c a1 00 00 00 01
0x01c0   01 01 0f fe ff ff c1 3e-00 00 55 8a 1f 01 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

This as we have seen above points to the EBR (16,065) in the Extended Partition.

**Block 16065 – The (first) Extended Partition Boot Record (EBR)**

```
LBN 16065

******
0x01b0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 01
0x01c0   01 01 07 fe 7f fe 3f 00-00 00 3f 04 7d 00 00 00
0x01d0   41 ff 05 fe ff fc 7e 04-7d 00 7e 04 7d 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 aa
```

As before, the first logical drive is at offset 63 in this (first extended partition). The second logical drive is found in the (second) extended partition at offset 0x007D047E (8,193,150) into the first extended partition. This offset has to be added to the offset of the EBR (16,065). This means that the EBR for the second extended partition is at offset 8,209,215.

**Block 8,209,215– The (second) Extended Partition Boot Record (EBR)**

```
LBN 9209215

0x01b0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 01
0x01c0   41 ff 07 fe ff fc 3f 00-00 00 3f 04 7d 00 00 00
0x01d0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01e0   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0x01f0   00 00 00 00 00 00 00 00-00 00 00 00 00 55 aa
```

Here is the definition of the second logical drive. It is at offset 0x0000003F (63) in the partition. The actual offset is 8,209,278 (63 + 8,209,215) blocks.

**Block 8,209,278– The second logical drive**

```
LBN 8209278

0x0000   eb 52 90 4e 54 46 53 20-20 20 20 00 02 08 00 00   ÙRÉNTFS    .....
0x0010   00 00 00 00 00 f8 00 00-3f 00 ff 00 3f 00 00 00   .....°..?. .?...
0x0020   00 00 00 00 80 00 80 00-3e 04 7d 00 00 00 00 00   ....Ç.Ç.>.}.....
0x0030   00 00 04 00 00 00 00 00-43 d0 07 00 00 00 00 00   ........CÐ......
0x0040   f6 00 00 00 01 00 00 00-4d 1a 99 5c 2a 99 5c 82   ÷.......M?Ö\*Ö\é
0x0050   00 00 00 00 fa 33 c0 8e-d0 bc 00 7c fb b8 c0 07   ....·3+Äð+.|¹©+.
0x0060   8e d8 e8 16 00 b8 00 0d-8e c0 33 db c6 06 0e 00   ÄÏÞ?.©..Ä+3¦ã...
0x0070   10 e8 53 00 68 00 0d 68-6a 02 cb 8a 16 24 00 b4   ?ÞS.h..hj.-è?$.¦
0x0080   08 cd 13 73 05 b9 ff ff-8a f1 66 0f b6 c6 40 66   .-?s.¦  è±f.Âã@f
0x0090   0f b6 d1 80 e2 3f f7 e2-86 cd c0 ed 06 41 66 0f   .ÂÐÇÔ?¸Ôå-+Ý.Af.
0x00a0   b7 c9 66 f7 e1 66 a3 20-00 c3 b4 41 bb aa 55 8a   À+f¸ßfú .+¦A+¬Uè
0x00b0   16 24 00 cd 13 72 0f 81-fb 55 aa 75 09 f6 c1 01   ?$.-?r.ü¹U¬u.÷-.
0x00c0   74 04 fe 06 14 00 c3 66-60 1e 06 66 a1 10 00 66   t.¦.¶.+f`?.fí?.f
0x00d0   03 06 1c 00 66 3b 06 20-00 0f 82 3a 00 1e 66 6a   ..?.f;. ..é:.?fj
0x00e0   00 66 50 06 53 66 68 10-00 01 00 80 3e 14 00 00   .fP.Sfh?...Ç>¶..
0x00f0   0f 85 0c 00 e8 b3 ff 80-3e 14 00 00 0f 84 61 00   .à..Þ¦ Ç>¶...äa.
0x0100   b4 42 8a 16 24 00 16 1f-8b f4 cd 13 66 58 5b 07   ¦Bè?$.??ï¶-?fX[.
0x0110   66 58 66 58 1f eb 2d 66-33 d2 66 0f b7 0e 18 00   fXfX?Ù-f3Êf.À.?.
0x0120   66 f7 f1 fe c2 8a ca 66-8b d0 66 c1 ea 10 f7 36   f¸±¦-è-fïðf-Û?¸6
0x0130   1a 00 86 d6 8a 16 24 00-8a e8 c0 e4 06 0a cc b8   ?.åÍè?$.èÞ+õ..¦©
0x0140   01 02 cd 13 0f 82 19 00-8c c0 05 20 00 8e c0 66   ..-?.é?.î+. .Ä+f
0x0150   ff 06 10 00 ff 0e 0e 00-0f 85 6f ff 07 1f 66 61    .?. ....ào .?fa
0x0160   c3 a0 f8 01 e8 09 00 a0-fb 01 e8 03 00 fb eb fe   +á°.Þ..á¹.Þ..¹Ù¦
0x0170   b4 01 8b f0 ac 3c 00 74-09 b4 0e bb 07 00 cd 10   ¦.ï-¼<.t.¦.+..-?
0x0180   eb f2 c3 0d 0a 41 20 64-69 73 6b 20 72 65 61 64   Ù=+..A disk read
0x0190   20 65 72 72 6f 72 20 6f-63 63 75 72 72 65 64 00    error occurred.
0x01a0   0d 0a 4e 54 4c 44 52 20-69 73 20 6d 69 73 73 69   ..NTLDR is missi
0x01b0   6e 67 00 0d 0a 4e 54 4c-44 52 20 69 73 20 63 6f   ng...NTLDR is co
0x01c0   6d 70 72 65 73 73 65 64-00 0d 0a 50 72 65 73 73   mpressed...Press
0x01d0   20 43 74 72 6c 2b 41 6c-74 2b 44 65 6c 20 74 6f    Ctrl+Alt+Del to
0x01e0   20 72 65 73 74 61 72 74-0d 0a 00 00 00 00 00 00    restart........
0x01f0   00 00 00 00 00 00 00 00-83 a0 b3 c9 00 00 55 aa   ........âá¦+..U¬
```

This is the NTFS boot record for the second logical drive. Unfortunately, this address is not divisible by 128 and therefore is not aligned.

## *Step 3: Convert to Dynamic and Check the Alignment*

After conversion to Dynamic, `dmdiag -v` gives:

```
Sub Disk  Rel Sec    Tot Sec  Tot Size  Plex        Vol Type  Col/Ord  DevName    State
========  =======    =======  ========  ====        ========  =======  =========  ======
Disk1-01  16128      8193087  18874368  Volume1-01  Simple    1/1  Harddisk1  ONLINE
Disk1-02  8209278    8193087  18874368  Volume2-01  Simple    1/1  Harddisk1  ONLINE
LDM-DATA  18872320   2048
```

The first volume is still at block 16,128 and the second at 8,209,278. This can be verified with `secinspect` or `dskprobe2` so it is possible to have an aligned Dynamic Volume that was created from a logical drive in an extended partition. Note, of course, in this example the second logical drive was *not* originally aligned, and therefore is still *not* aligned after conversion. The LDM metadata *is* aligned.

If subsequent volumes are created, these volumes are *not* aligned. This is as a consequence of the second logical drive being misaligned. As shown next, the third volume starts at offset 16,402,365 and is *not* aligned.

```
Sub Disk   Rel Sec    Tot Sec   Tot Size   Plex         Vol Type   Col/Ord   DevName     State
========   =======    =======   ========   ====         ========   =======   =========   ======
Disk1-01   16128      8193087   18874368   Volume1-01   Simple     1/1   Harddisk1   ONLINE
Disk1-02   8209278    8193087   18874368   Volume2-01   Simple     1/1   Harddisk1   ONLINE
Disk1-03   16402365   2048000   18874368   Volume3-01   Simple     1/1   Harddisk1   ONLINE
LDM-DATA   18872320   2048
```

If the last two volumes are deleted (leaving just the one aligned volume), and creating of another Simple Dynamic Volume, the following layout is output by `dmdiag -v`:

```
Sub Disk   Rel Sec    Tot Sec   Tot Size   Plex         Vol Type   Col/Ord   DevName     State
========   =======    =======   ========   ====         ========   =======   =========   ======
Disk1-01   16128      8193087   18874368   Volume1-01   Simple     1/1   Harddisk1   ONLINE
Disk1-02   8209215    8192000   18874368   Volume2-01   Simple     1/1   Harddisk1   ONLINE
LDM-DATA   18872320   2048
```

The new volume at offset 8,209,215 is *not* aligned either. This is because the first volume's size is *not* a multiple of 128 blocks and Windows creates the second volume immediately after the first volume.

## Conclusion

You can align a volume on a Windows Basic Disk by creating a logical drive in an extended partition. The steps are:

1. Ensure the disk is a Basic Disk. Delete all volumes and revert to Basic Disk if necessary.
2. Using the LDM GUI or `diskpart`, create an extended partition.
3. Create a logical drive in the extended partition. This logical drive *will* be aligned to 128 blocks.
4. If this logical drive is upgraded to a Dynamic Volume, it *will* be aligned.
5. Subsequent volumes are *not* aligned under any circumstances.
6. You can only have one Simple Volume aligned using this method

Note that the previous will only be true if Windows determines that the disk has a drive geometry of 255 heads and 63 sectors per track. If any other drive geometry is used, the logical drive will almost certainly be misaligned.

# Summary

The following is a summary of the guidelines for using `diskpar` to align partitions on Basic Disks and Dynamic Disks under both Windows 2000 and Windows 2003.

### Windows 2000 Basic Disks

- Use `diskpar` with an offset of 128 (Symmetrix) or 128[2] (CLARiiON) and the desired partition size.
- You can create one more aligned partition with the GUI only. If you use `diskpart` to create a subsequent partition, the partition will not be aligned.

### Windows 2003 Basic Disks

- Use `diskpar` or `diskpart` with an offset of 128 (Symmetrix) or 128 (CLARiiON) and the desired partition size.
- You cannot create any more aligned partitions with either the GUI or `diskpart`. All subsequent partitions created are misaligned.

### Windows 2000 Dynamic Disks

- Use `diskpar` with an offset of 128 (Symmetrix) or 128 (CLARiiON) and a partition size of 8 MB.
- Convert the Basic Disk to a Dynamic Disk.
- All subsequent volumes created with the GUI or `diskpart` are aligned.

### Windows 2003 Dynamic Disks

- Use `diskpar` with an offset of 128 (Symmetrix) or 128 (CLARiiON) and a partition size of 16 MB.
- Convert the Basic Disk to a Dynamic Disk.
- All subsequent volumes created with the GUI or `diskpart` are aligned.
- You cannot use `diskpart` to align volumes on a Windows 2003 SP1 Dynamic Disk.

### Creating a Logical Drive inside an Extended Partition

- You can create one logical drive inside an extended partition.
- You can upgrade this to a Dynamic Disk.
- You cannot add any more aligned volumes irrespective of whether the disk is Basic or Dynamic.

[2] This is the default CLARiiON element size. You must check the CLARiiON LUN properties to confirm this. Read on for details.

# Appendix A: Master Boot Record (MBR) Contents

The MBR is always found in the very first block on a disk (Block 0). You can use the `secinspect` or `dskprobe2` tool to look at the MBR.

Following is a typical MBR. The colored areas show the various sections of the block.

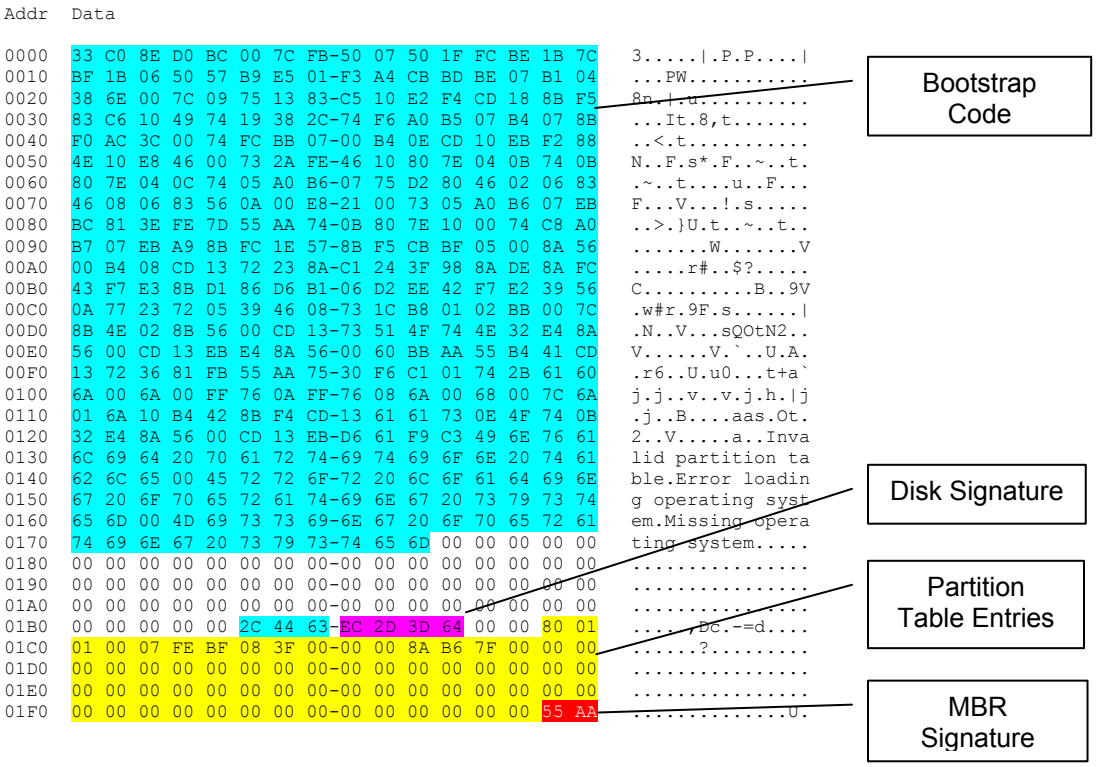



**Figure 10. Typical Master Boot Record**

## *Bootstrap Code*

When you boot the computer, the BIOS reads this MBR into location 0x7C00 in memory and starts executing this bootstrap code from the address 0x7C00. First, the code copies itself to location 0x0600 because 0x7C00 is required when the partition's boot sector is read in. (It actually skips copying the first 0x001A bytes of the bootstrap code because that is the code that does the copying and it is not needed after the actual copy.)

The bootstrap code (now running at location 0x61B) then looks for a partition entry with the active flag set, loads the first sector of that partition into memory at location 0x7C00, and then executes the code. This partition boot code is different for each operating system.

## *Disk Signature*

Every disk in Windows must have a disk signature to be useable. This is not strictly true—obviously if the disk was not useable, you would never be able to write a signature to it. Disk tools such as `secinspect` and `dskprobe2` can read or write to a disk, and in reality so can any application. For the disk to be treated as anything other than a raw disk though, the disk must have a signature on it.

These signatures must be unique within a system and duplicate signatures can cause problems. Duplicate signatures are typically seen when more than one path is available to a disk but no multipath software (such as EMC PowerPath®) is installed. Windows 2003 is more tolerant of multipathed disks. If Windows 2003 discovers a disk with the same signature as a previously discovered disk, Windows 2003 will change the signature of the second disk if it is determined that this is *not* a second path to the first disk. Note that the change is only made to the Windows Registry; the disk itself is not actually changed.

## *Partition Table Entries*

A disk can have up to four primary partitions defined and each entry is 16 bytes long. The partition table area in the previous example is interpreted as follows:

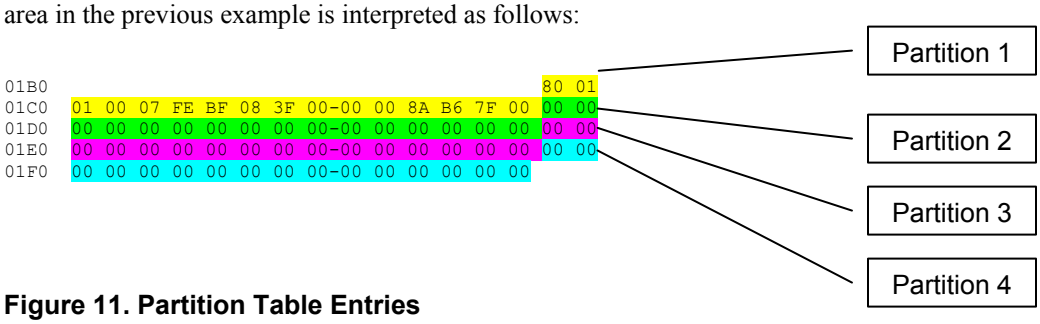



**Figure 11. Partition Table Entries**

The only fields in a partition table entry that are relevant in these days of 32-bit operating systems and large (greater than 7.8 GB) disks are:

- Active Flag
- System ID (Partition Type)
- Partition Offset
- Partition Length

You can usually ignore the rest, but there are some conditions where Windows will check the Starting and Ending Cylinder, and Head and Sector values.

The following identifies the components of a partition table entry (e.g., Partition 1 in Figure 11):

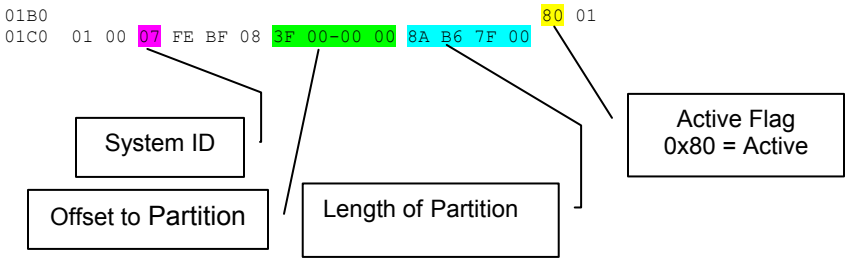



**Figure 12. Components of Partition Table Entry**

### Active Flag

This byte has either a value of 0 or is set to 0x80 to indicate that this partition is the bootable partition.

Setting the active flag on a disk that cannot or will not be booted from is usually safe and can easily be inadvertently done using the Disk Management GUI. Unfortunately, the GUI offers no method of turning it off. The worst thing that can happen is that the system will hang on booting if this disk is booted. One other interaction is that DRU will not migrate a disk with the active flag set, so if it is accidentally set, DRU will fail.

## System ID

This shows the partition type. Common values are:

0x05 Extended Partition
0x07 NTFS
0x42 Windows 2000/2003 Dynamic
0xEE Windows 2003 GPT Partition

## Offset to Partition

This is the zero-based offset (in blocks) to the partition on the disk. The default value is almost always 63 (0x3F), and you must adjust this value using `diskpar`.

## Length of Partition

This is the size of the partition in blocks.

## MBR Signature

The last two bytes of the MBR are always 0x55AA.