

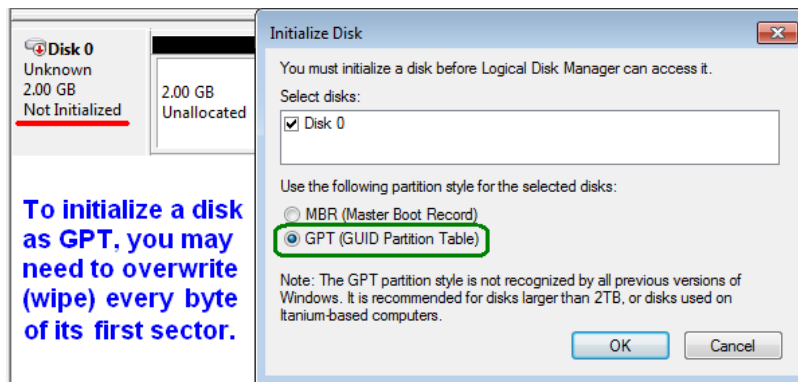
An Examination of *The* Microsoft® Windows™ 7 and 8 GPT 'Protective' MBR and EFI Partitions

Web Presentation and Text are Copyright©2013 by Daniel B. Sedory
NOT to be reproduced in any form without Permission of the Author !

- [Introduction](#)
- [The GPT Protective MBR and Partition Table](#)
- [The GPT Header](#)
 - Sample "Hex Dump" of GPT Header bytes
 - [Table of GPT Header Fields](#)
- [Calculating the CRC32 Checksums](#)
-

Introduction

Here we examine the structure created by the Windows™ 7 (64-bit) "Logical Disk Manager" when GPT partitioning is chosen. The major reason for the invention of the **GPT** ("**GUID Partition Table**") was the limitation of the MBR Partition Table to a maximum partition size of 4,294,967,295 sectors (about 2.2 TB). However, the *minimum* capacity for a GPT partitioned drive is *far* less than 2 TB. In fact, we created our first GPT partitioned disk drive (in a virtual computer) with a capacity of only 2 GiB (after wiping every byte of the drive with the hex value 0x22, so we could know for sure if any sectors were zero-filled by the OS):



NOTE: The Windows 7 (or 8) **Logical Disk Manager** does **not** distinguish between disks partitioned as **MBR** or **GPT**; both disk types are simply listed as "Basic"! One quick way to determine if a disk is GPT partitioned is to use the **List Disk** command of the "**DISKPART**" tool under a Command Prompt (may require Administrator rights). Simply enter the commands shown in **GREEN** below. If any attached disk has been GPT partitioned, an asterisk (*) will appear under the "Gpt" column; as shown in **YELLOW** for Disk 0 below:

```

Administrator: Command Prompt

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>diskpart

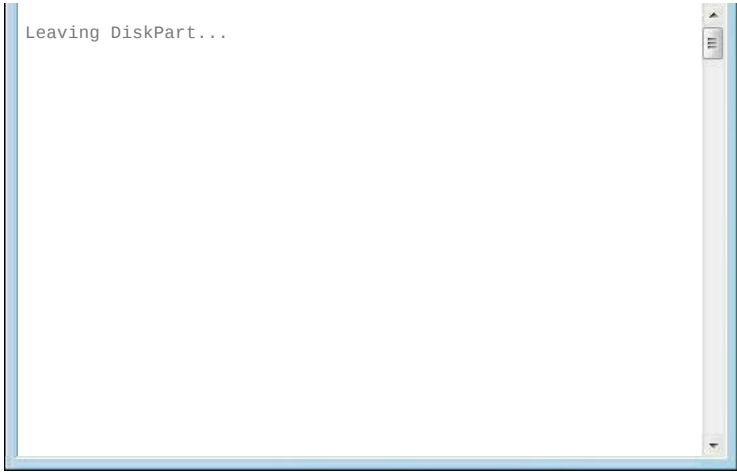
Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: MyPC

DISKPART> list disk

   Disk ###  Status         Size      Free      Dyn  Gpt
   -----  -
   Disk 0    Online         2047 MB   1920 KB          *
   Disk 1    Online         24 GB              0 B

DISKPART> exit

```



Of course, using a disk editor, such as the free [HxD](#) or the demo version of [WinHex](#) (which, along with [TestDisk](#), also understands GPT drives) is the best way to examine the actual contents of your disk drive(s) and see all the details presented below.

The GPT Protective MBR and Partition Table

When a disk is initialized as **GPT** by Windows 7 (or Windows 8), the first sector (Absolute Sector 0) still contains the same boot code as a [Windows 7 MBR](#) initialized disk drive, but with this exception: The *NT Disk Signature* is set to zero ("00 00 00 00" as shown in Figure 1 below). Note, however, that connecting such a drive to a Windows XP (and perhaps others) PC, will result in an *NT Disk Signature* being assigned to the drive. The partition table contains only a **single "GPT Protective" entry** which in all cases is set to the maximum 32-bit limitation (even though a drive may have far less than a 2.2 TB capacity). The "GPT Protective MBR Sector" has **exactly the same contents for all GPT disk drives created by the Windows 7 (or 8) OS**. But, note: This does **not** follow the UEFI Specification, which states that the "*SizeInLBA*" should be "set to the size of the disk minus one" *if it's not too large to be represented*.^[1] (GPT drives partitioned under various **Linux** and **Apple® Mac** OS systems do follow the UEFI Specification in this regard.)

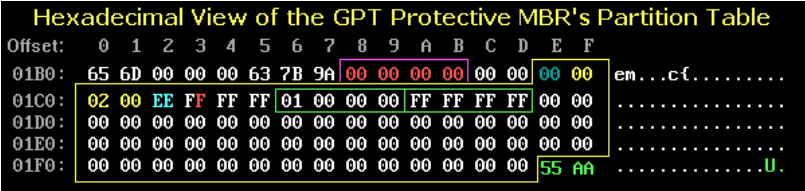


Figure 1.

Figure 2 shows how to interpret this 16-byte [Partition Table entry](#) in Figure 1: The first byte ("00") at offset 1BEh means **not** bootable (but this is only to ensure a legacy BIOS or OS will not attempt to boot from this drive; [EFI/UEFI](#) BIOS do not use this byte!), the next three bytes ("00 02 00") indicate the partition starts at CHS (0,0,2) or Absolute **Sector 1** (this is where the **GPT Header** is located). The Partition Type byte at offset 1C2h ("EE") indicates this is a GPT partitioned disk. The next three bytes hold the CHS Ending values which were only used if the partition didn't end beyond 16,450,560 sectors (or about 8.4 GB). Otherwise, it was agreed to fill these bytes with "FE FF FF" (CHS values **1023, 254** and 63, for a total of: **1024** cylinders, **255** heads and 63 sectors); see our note on the MS-DOS [Head Count Error](#) for the reason all BIOS companies standardized the head count at 255 instead of 256. **But** here we have an 'FF' in every byte: **FE FF FF**". Though used only to ensure no legacy utility or OS will find any unpartitioned space on the disk, by always filling this field with 0xFFFFFFFF, the Windows 7 (or 8) OS is **not** following the UEFI Specification which states the *EndingCHS* should be "set to the CHS address of the last logical block on the disk" *unless it's too large to be represented*.^[2] The next 4 bytes in the first GREEN box ("01 00 00 00") indicate the number of sectors which precede the partition (sometimes called "Relative Sectors"); so, one sector: The MBR. And the last 4 bytes ("FF FF FF FF") indicate a partition size of 4,294,967,295 sectors. As we mentioned above, Windows 7 *always* fills this field with 0xFFFFFFFF, even though the UEFI Specification states this should be "set to the size of the disk minus one" for drives under 2.2 TB.^[3]

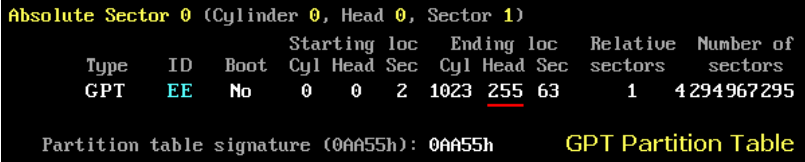


Figure 2.

The GPT Header

The GPT Header immediately follows the MBR in Absolute (i.e., LBA) Sector 1; the second Sector on the drive. Figure 3 shows a *Disk Editor View* (or 'Hex Dump') of an example GPT Header of only its 92 bytes of data:

The GPT Header data is found only in the first 92 bytes of this zero-padded sector.																
Absolute (LBA) Sector 1 (Cylinder 0, Head 0, Sector 2)																
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

0000	45	46	49	20	50	41	52	54	00	00	01	00	5C	00	00	00	EFI PART....\...
0010	C8	03	40	5B	00	00	00	00	01	00	00	00	00	00	00	00	..@[.....
0020	EF	FF	3F	00	00	00	00	00	22	00	00	00	00	00	00	00	..?....."
0030	CE	FF	3F	00	00	00	00	00	5E	86	90	EF	D0	30	03	46	..?.....^.....0.F
0040	99	3D	54	6E	B0	E7	1B	0D	02	00	00	00	00	00	00	00	..Tn.....
0050	80	00	00	00	80	00	00	00	1E	C5	0F	B7				
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 3.

The GPT Header always begins with the 8-byte EFI "Signature" string: "45 46 49 20 50 41 52 54" (ASCII: "EFI PART"); declared in the UEFI Specification as the 64-bit Hex constant: "0x5452415020494645". The following table explains every byte in this Header:

Specifications for the GPT Header Sector		
Offset (Hex)	Length (bytes)	Contents
000	8	Signature: Identifies this as an EFI-compatible GPT Header; must contain the ASCII string: "EFI PART"
008	4	Revision: Revision number for this header. This revision is not related to the UEFI Specification version. The current GPT Header version is still 1.00, so 0x00010000 (or "00 00 01 00") is the correct value.
00C	4	HeaderSize: Size in bytes; usually 92 (5Ch), the field being filled with: "5C 00 00 00". The UEFI Spec states this "must be greater than 92" (and should be the logical block size). But every drive we've examined does not follow the UEFI Specification here ^[4] .
010	4	HeaderCRC32: CRC32 of the GPT Header, with this field zeroed during calculation. (In our example above, this is "C8 03 40 5B". See below: Calculating the CRC32 Checksums.)
014	4	Reserved; must be set to zero ("00 00 00 00").
018	8	MyLBA: LBA location of this GPT Header. (Most often 0x0000000000000001.)
020	8	AlternateLBA: LBA address of the <i>alternate</i> GPT Header (i.e., the location of the backup copy of this header). Note: That EFI Header will have its own 'MyLBA' location and also its own CRC32 checksum, and if fact, will point to where this EFI Header is located in its 'AlternateLBA' field. [In our example above, the "EF FF 3F 00 00 00 00 00" points to Sector 4194287.]
028	8	FirstUsableLBA: First usable LBA Sector for any partition; equal to: 1 + the last LBA Sector of the Primary GUID Partition Table. Most often Sector 34 (0x0000000000000022), which is "22 00 00 00 00 00 00 00" in our example above.
030	8	LastUsableLBA: Last usable LBA Sector for any partitions; equal to: The first LBA Sector of the Secondary Partition Table - 1. In our example above, this is Sector 4194254 ("CE FF 3F 00 00 00 00 00").
038	16	DiskGUID: Disk GUID (Globally Unique Identifier; also referred as UUID, UniversallyUID, on UNIXes) for this physical disk drive. In our example above this is the whole 16 byte string of "5E 86 90 EF D0 30 03 46 99 3D 54 6E B0 E7 1B 0D" (or 0x0d1be7b06e543d99460330d0ef90865e).
048	8	PartitionEntryLBA: Starting LBA of the array of GUID Partition entries; most often Sector 2 for the Primary Partition. ("02 00 00 00 00 00 00 00" in our example above.)
050	4	NumberOfPartitionEntries: Number of Partition Entries in the GUID Partition Entry array. Most often 0x00000080 (which is 128).
054	4	SizeOfPartitionEntry: Size, in bytes, of each GUID Partition Entry. This field should be set to a value of: 128 x 2n where n is an integer greater than or equal to zero (thus, although this is usually set to 128, it may also be 256, 512, etc.) ^[5] . 0x00000080 (80 hex) = 128.
058	4	PartitionEntryArrayCRC32: CRC32 of the GUID Partition Array. ("Starts at <i>PartitionEntryLBA</i> and is computed over a byte length of <i>NumberOfPartitionEntries</i> * <i>SizeOfPartitionEntry</i> .") In our example above, this is "1E C5 0F B7" (or 0xb70fc51e).
05C	(BlockSize - 92)	Reserved; the remainder of the block is reserved by UEFI and must be padded with zero bytes.

Calculating the CRC32 Checksums

The CRC32 checksum for the example GPT Header in Figure 3 (hex offsets 00 through 5B or 0x5C = 92 bytes; with offsets 10 - 13 zeroed out) is: 0x5b4003c8 ("C8 03 40 5B" being the *on disk little-endian* representation of this hexadecimal number). This can be verified by using [HxD's "Analysis" -> "Checksums..."](#) tool as follows: Temporarily change the bytes at hex offsets 10 - 13 to zero bytes (shown in **RED** in Figure 4 below), select all 92 bytes (as indicated by the BLUE highlighting), then choose to perform a "CRC-32" checksum (**not** 'Checksum-32') operation on the selected bytes:

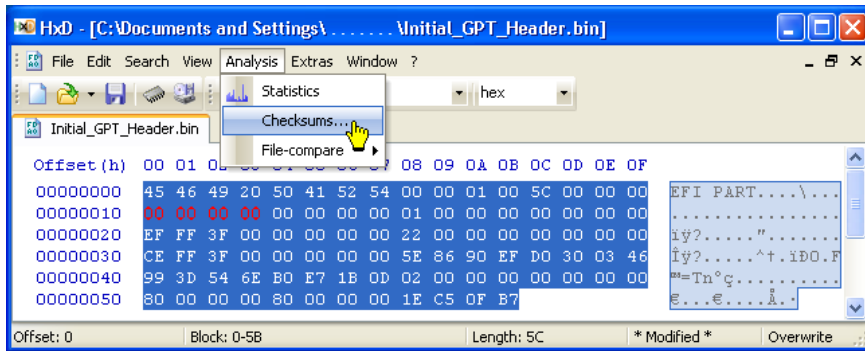


Figure 4.

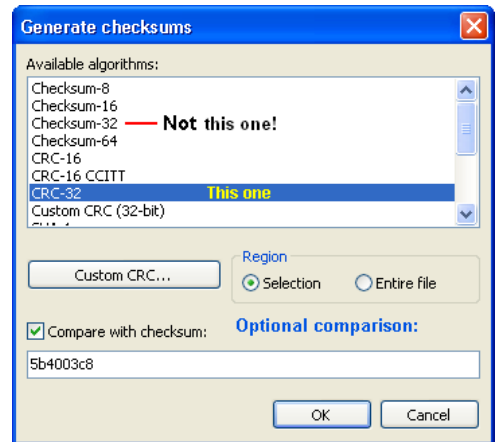


Figure 5.

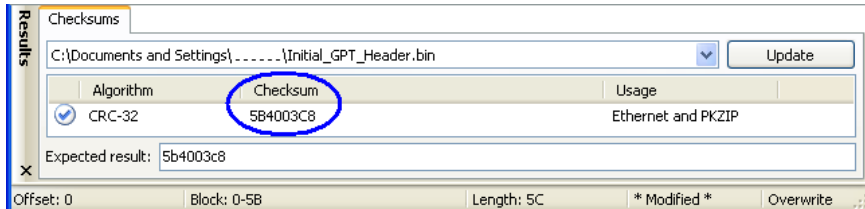


Figure 6.

The remainder of this page is currently a work in progress. We must address the layout of the GUID Partition Array itself and its (usually) 128 possible entries. There are also 'backup copies' of this array and the GPT Header at the end of each GPT partitioned disk drive which we'll show for a 3 TB example drive.

Footnotes

¹[\[Return to Text\]](#) Reference and full quote: *Unified Extensible Firmware Interface Specification, Version 2.3.1, Errata C, June 27, 2012*, which states in Chapter 5, GUID Partition Table (GPT) Disk Layout, Section 5.2.3, Protective MBR, Table 15, '**SizeInLBA**', on page 100: "Set to the size of the disk minus one. Set to 0xFFFFFFFF if the size of the disk is too large to be represented in this field." Since Microsoft uses the same entry for drives smaller than 2.2 TB as it does for those over 2.2 TB, they are not following the UEFI Specification for **SizeInLBA**.

You can download the  [UEFI Specifications here](#).

²[\[Return to Text\]](#) Reference and full quote: *Unified Extensible Firmware Interface Specification, Version 2.3.1, Errata C, June 27, 2012*, which states in Chapter 5, GUID Partition Table (GPT) Disk Layout, Section 5.2.3, Protective MBR, Table 15, '**EndingCHS**', on page 100: "Set to the CHS address of the last logical block on the disk. Set to 0xFFFFFFFF if it is not possible to represent the value in this field." Since Microsoft uses the same exact CHS values for drives smaller than 2.2 TB as it does for those over 2.2 TB, they are not following the UEFI Specification for the **EndingCHS** values.

³[\[Return to Text\]](#) See **Footnote # 1**.

⁴[\[Return to Text\]](#) The *Unified Extensible Firmware Interface Specification, Version 2.3.1, Errata C, June 27, 2012*, states on page 104, in Table 16: "Size in bytes of the GPT Header. The **HeaderSize** must be greater than 92 and must be less than or equal to the logical block size."

NOTE: After contacting the UEFI organization, they assured us that the Specification does in fact mean a value **larger than** 92 bytes should be used; and that it should be equal to the logical block size (which would be 512 bytes in our example where the GPT Partition Array begins at LBA 2). **However, every GPT partitioned drive** we've examined, whether it was partitioned by a Microsoft Windows 7 or 8 OS, a **Linux** utility such as **GParted** or even an **Apple Mac OSX** machine, has **exactly 92 bytes** ("5C 00 00 00") in this field. Presumably, this is why one program even states of the Header Size field: "should be 92" (according to *WinHex* in their latest demo version downloaded on 3 JAN 2013). If you know of any utility or OS that actually follows the UEFI Specification for this field, please let us know.

⁵[\[Return to Text\]](#) Some previous versions of the UEFI Specification stated that any "multiple of 8" could be used (*Unified Extensible Firmware Interface Specification, Version 2.3.1, Errata C, June 27, 2012*, page 105, Table 16).

*Last Update: **January 7, 2013** (07.01.2013).*

You can write to me using this: [contact page](#) (opens in a new window).

 [MBR and Boot Records Index](#)

 [The Starman's Realm Index Page](#)