NTFS.com NTFS: Hard Disk Drive Basics

NTFS General Info Data Recovery Software FAQ Contacts NTFS Forum Site Map

NTFS General Information > File Storage Hardware and Disk Organization > GUID Partition Table (GPT)

GUID Partition Table (GPT)

GPT - Technical Documentation

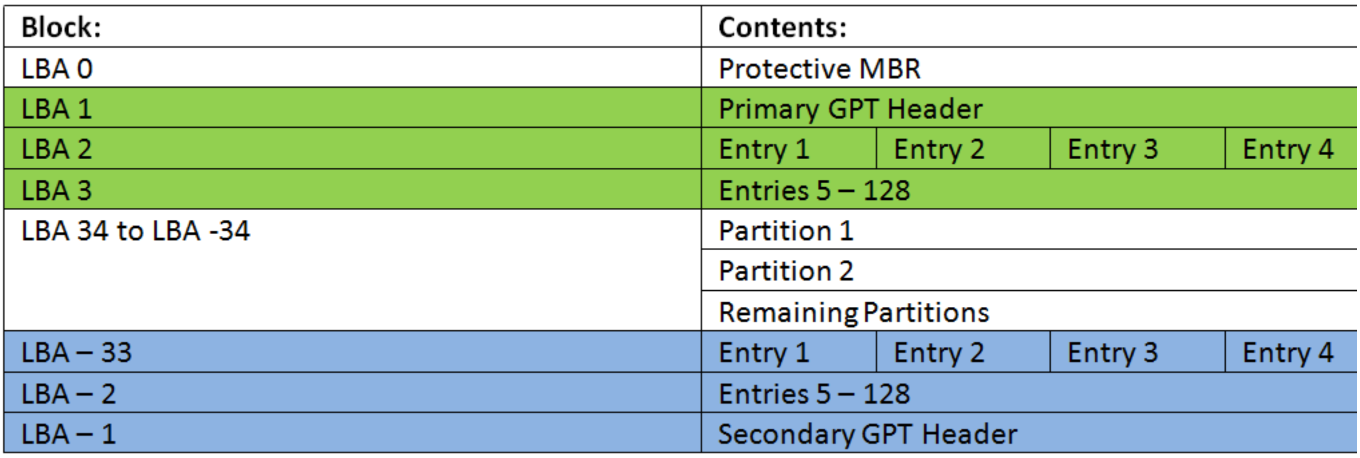
Introduction:

The GUID Partition Table, known as the GPT, is a popular disk partitioning scheme used across most operating systems, including Windows and Unix-class operating systems such as Mac OS X.

It was introduced by Intel in the late 1990's and has since become the standard layout of the partition table on a physical hard disk. It is a successor to many partition tables, such as MBR and APM, overcoming their limitations of using 32 bits for logical block addresses and a standard block size of 512 bytes.

These limitations restrict the disk size of the machine to 2.2TB. With the rate of increase in hard drive capacity in computers, this limitation is significant enough for switching most of the older partitioning schemes to the GPT, which allocates 64 bits for logical block addresses, translating to 9.4 ZB(zeta-bytes, or $9.4*10^{21}$ bytes) of potential capacity. With the entire World Wide Web being estimated to be about 0.5 ZB in size, the GPT partitioning system has the potential to stay around for a long time.

The GPT:
Figure 1 gives a scheme of the GPT:



Block:	Contents:			
LBA 0	Protective MBR			
LBA 1	Primary GPT Header			
LBA 2	Entry 1	Entry 2	Entry 3	Entry 4
LBA 3	Entries 5 – 128			
LBA 34 to LBA -34	Partition 1			
	Partition 2			
	Remaining Partitions			
LBA – 33	Entry 1	Entry 2	Entry 3	Entry 4
LBA – 2	Entries 5 – 128			
LBA – 1	Secondary GPT Header			

Note: The green block represents the Primary GPT, the blue block represents the secondary GPT

Figure 1.

The GPT, or the GUID Partition Table, is the standard format of partitioning tables on a physical hard disk. It was introduced as part of the EFI, or Extensible Firmware Interface, standard, created by Intel to replace the outlived BIOS, one of the last relics of the ancient IBM PC. The EFI uses the GPT where BIOS used the MBR.

Unlike the MBR, which starts with an executable program, called to identify and load the active sector, the GPT uses the wider range of possibilities of the EFI to start up these processes. However, the MBR is present at the beginning of the disk, in block LBA0, for protective and compatibility purposes. Strictly speaking, the GPT starts up from the Partition Table Header.

The GPT uses a modern system of Logical Block Addressing (LBA) rather than the outdated CHS (Cylinder-Head-Sector) method, used in MBR. Passed down from its predecessor, block LBA 0 contains MBR, with the Primary GPT Header in LBA 1. Following the GPT Header is the partition table itself. In the 64-bit version of Microsoft Windows NT, the GPT reserves the first 32 sectors, so that the first used sector on every hard drive is LBA 34.

That aside, the GPT provides data duplication; the header and partition table are stored both at the start and the end of the disk.

Theoretically, the GPT allows the partitioning of disks up to 9.4 ZB, while MBR works only up to 2.2 TB.

Protective MBR Block (LBA 0):

The main purpose of inserting the MBR at the start of disk is strictly for protection. MBR-oriented disk utilities have the possibility of not recognizing or even writing over top of the GPT disks. To prevent this, the entire GPT disk is labeled as one partition. The System ID of this partitoin is established as 0xEE, indicating the implementation of the GPT.

After this, the EFI ignores the MBR. Several 32-bit operating systems, not compatible with reading disks using the GPT scheme, still recognize the System ID and recognize the GPT disk as inaccessible, preventing the overwriting of the GPT disk.

Older operating systems often recognize the disk as containing a single partition of an unknown type, without free space; and therefore, as a rule of thumb, refuse to modify this type of disk unless the user specifically demands that this be done. This helps prevent the accidental deletion of contents on the GPT disk.

Primary GPT Header (LBA 1):

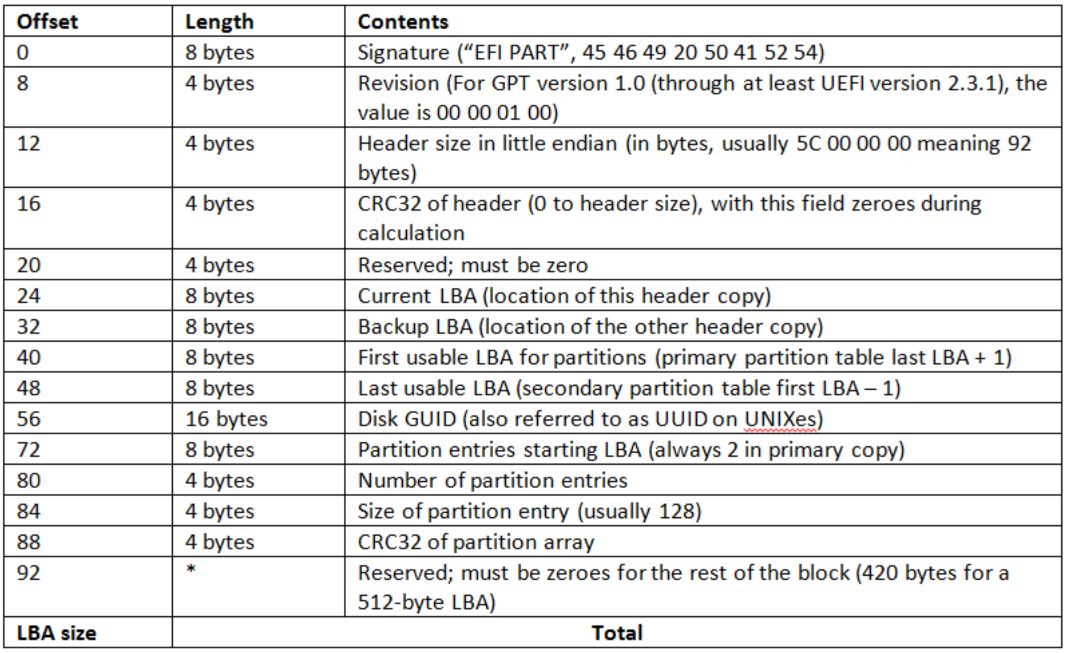
The GPT Header indicates the usable logical blocks on the disk. Also, it indicates the number and size records of the partitions, forming the partition table on a machine that contains the 64-bit version of Microsoft Windows Server 2003, 128 records of partition information are reserved, with each record being 128 bytes. This way, it is possible to create 128 partitions on the disk.

The GUID (Globally Unique Identifier) of the disk contains the header. Inside it is its' size and location (always block LBA 1), as well as the size and location of the secondary (emergency) header and partition table, always found in the last sectors of the disk.

As an important note, it also contains the Cyclic Redundancy Check CRC32 for itself and the partition table. It is run through an EFI process once the machine is started up. Because of the checksum, it is impossible to use disk Hex Editors to edit raw GPT data.

Any modifications to the GPT will change the checksum, after which the EFI will rewrite the primary GPT with the secondary one. If both of the GPTs hold an invalid checksum, access to the disk will become impossible.

The following table shows the GPT Header Format:



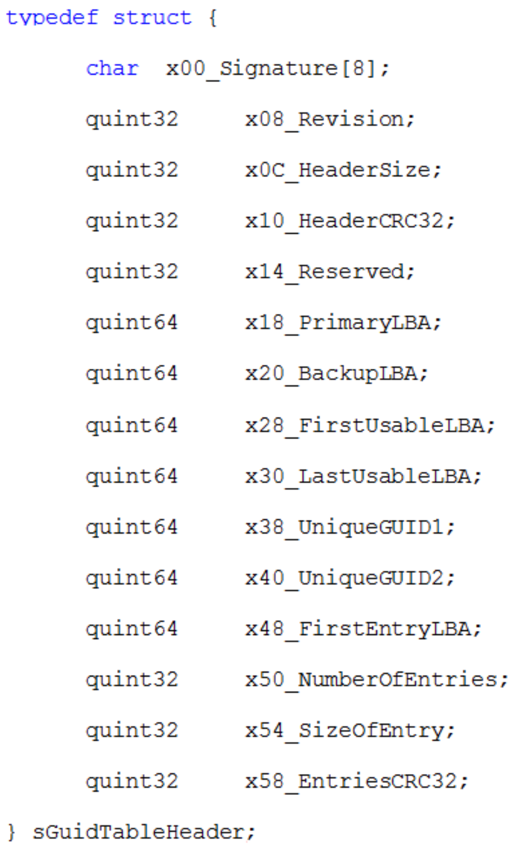
Offset	Length	Contents
0	8 bytes	Signature ("EFI PART", 45 46 49 20 50 41 52 54)
8	4 bytes	Revision (For GPT version 1.0 (through at least UEFI version 2.3.1), the value is 00 00 01 00)
12	4 bytes	Header size in little endian (in bytes, usually 5C 00 00 00 meaning 92 bytes)
16	4 bytes	CRC32 of header (0 to header size), with this field zeroes during calculation
20	4 bytes	Reserved; must be zero
24	8 bytes	Current LBA (location of this header copy)
32	8 bytes	Backup LBA (location of the other header copy)
40	8 bytes	First usable LBA for partitions (primary partition table last LBA + 1)
48	8 bytes	Last usable LBA (secondary partition table first LBA – 1)
56	16 bytes	Disk GUID (also referred to as UUID on UNIXes)
72	8 bytes	Partition entries starting LBA (always 2 in primary copy)
80	4 bytes	Number of partition entries
84	4 bytes	Size of partition entry (usually 128)
88	4 bytes	CRC32 of partition array
92	*	Reserved; must be zeroes for the rest of the block (420 bytes for a 512-byte LBA)
LBA size	**Total**	

Partition Entries (LBA 2-33):

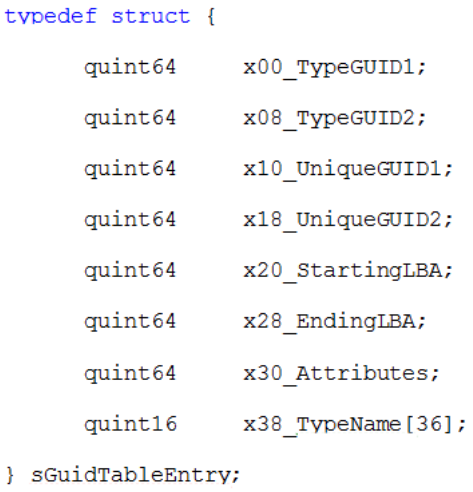
Partition entries are done simple and are done with equally incremented addresses. The first 16 bytes are used to define the GUID partition type. For example, the GUID EFI system appears as: "C12A7328-F81F-11D2-BA4B-00A0C9 3EC93B", with the next 16 bytes containing a GUID that is unique to the partition. After that, there is the record of the start and end of the 64-bit LBA, if applicable. The rest of the space is dedicated to the names and attributes of the partition.

C++ explanation:

The following is an example of C++ structures that are used with the GPT:



```
typedef struct {
        char  x00_Signature[8];
        quint32     x08_Revision;
        quint32     x0C_HeaderSize;
        quint32     x10_HeaderCRC32;
        quint32     x14_Reserved;
        quint64     x18_PrimaryLBA;
        quint64     x20_BackupLBA;
        quint64     x28_FirstUsableLBA;
        quint64     x30_LastUsableLBA;
        quint64     x38_UniqueGUID1;
        quint64     x40_UniqueGUID2;
        quint64     x48_FirstEntryLBA;
        quint32     x50_NumberOfEntries;
        quint32     x54_SizeOfEntry;
        quint32     x58_EntriesCRC32;
} sGuidTableHeader;
```



```
typedef struct {
        quint64     x00_TypeGUID1;
        quint64     x08_TypeGUID2;
        quint64     x10_UniqueGUID1;
        quint64     x18_UniqueGUID2;
        quint64     x20_StartingLBA;
        quint64     x28_EndingLBA;
        quint64     x30_Attributes;
        quint16     x38_TypeName[36];
} sGuidTableEntry;
```

previous | contents | next

Data Recovery
UNFORMAT
Active@ UNERASER
Active@ UNDELETE
Active@ File Recovery
Active@ Partition Recovery
Active@ Password Changer

Disk Utilities
Active@ Boot Disk(Live CD)
Active@ Partition Manager
Active@ Hard Disk Monitor
NTFS Reader for DOS
Active@ Disk Editor
NTFS Recovery toolkit

Data Security
Active@ KillDisk
Active@ ERASER
Active@ ZDelete
Active@ ZDelete Network

Data Backup
Active@ Disk Image

Data CD/DVD
Active@ ISO File Manager
Active@ ISO Burner
Active@ Data Burner
Active@ DVD Eraser

Be with us Follow us Look for us Contact us