GUID Partition Table

The **GUID Partition Table** (**GPT**) is a standard for the layout of partition tables of a physical computer storage device, such as a hard disk drive or solid-state drive, using universally unique identifiers, which are also known as globally unique identifiers (GUIDs). Forming a part of the Unified Extensible Firmware Interface (UEFI) standard (Unified EFI Forum-proposed replacement for the PC BIOS), it is nevertheless also used for some BIOS systems, because of the limitations of master boot record (MBR) partition tables, which use 32 bits for logical block addressing (LBA) of traditional 512-byte disk sectors.

All modern personal computer operating systems support GPT. Some, including macOS and Microsoft Windows on the x86 architecture, support booting from GPT partitions only on systems with EFI firmware, but FreeBSD and most Linux distributions can boot from GPT partitions on systems with both legacy BIOS firmware interface and EFI.

Contents

History

Features

MBR variants

- Protective MBR (LBA 0)
- Hybrid MBR (LBA 0 + GPT)

Partition table header (LBA 1)

Partition entries (LBA 2–33)

Operating-system support

- UNIX and Unix-like systems
- Windows: 32-bit versions
- Windows: 64-bit versions

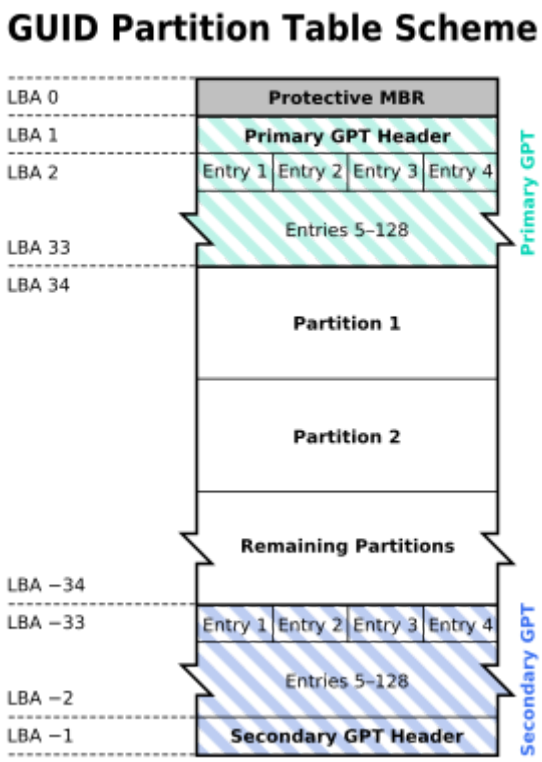
Partition type GUIDs

See also

Notes

References

External links



The layout of a disk with the GUID Partition Table. In this example, each logical block is 512 bytes in size and each entry has 128 bytes. The corresponding partition entries are assumed to be located in LBA 2–33. Negative LBA addresses indicate a position from the end of the volume, with −1 being the last addressable block.

History

The Master Boot Record (MBR) partitioning scheme, widely used since the early 1980s, imposed limitations for use of modern hardware. A major deficiency is the limited size of 32 bits for block addresses and related information. For hard disks with 512-byte sectors, the MBR partition table entries allow a maximum size of 2 TiB ($2^{32} \times 512$ bytes).[1]

In the late 1990s, Intel developed a new partition table format as part of what eventually became the Unified Extensible Firmware Interface (UEFI). As of 2010, the GUID Partition Table forms a subset of the UEFI specification.[2] GPT uses 64 bits for logical block addresses, allowing a maximum disk size of 2^{64} sectors. For disks with 512-byte sectors, the maximum size is 9.4 ZB (9.4×10^{21} bytes) or 8 ZiB (2^{64} sectors $\times$ 2^9 bytes per sector).[1][3]

Features

Like modern MBRs, GPTs use logical block addressing (LBA) in place of the historical cylinder-head-sector (CHS) addressing. The protective MBR is stored at LBA 0, the GPT header is in LBA 1, and the GPT header has a pointer to the partition table (*Partition Entry Array*), typically at LBA 2. The UEFI specification stipulates that a minimum of 16,384 bytes, regardless of sector size, are allocated for the Partition Entry Array.[4] Each entry has a size of 128 bytes. Thus, on a disk with 512-byte sectors, sector number 34 is the first usable sector on the disk.

Hard-disk manufacturers are transitioning new products to using 4,096-byte sectors. The first such drives continued to present 512-byte physical sectors to the operating system; degraded performance could result when the drive's physical 4-KiB sector boundaries did not coincide with the 4 KiB logical blocks, clusters and virtual memory pages common in many operating systems and file systems. This was a particular problem on write operations, when the drive is forced to perform two read-modify-write operations to satisfy a single misaligned 4 KiB write operation.[5]

MBR variants

Protective MBR (LBA 0)

For limited backward compatibility, the space of the legacy MBR is still reserved in the GPT specification, but it is now used in a way that prevents MBR-based disk utilities from misrecognizing and possibly overwriting GPT disks. This is referred to as a *protective MBR*.[3]

A single `partition type` of `EEh`, encompassing the entire GPT drive (where "entire" actually means as much of the drive as can be represented in an MBR), is indicated and identifies it as GPT. Operating systems and tools which cannot read GPT disks will generally recognize the disk as containing one partition of unknown type and no empty space, and will typically refuse to modify the disk unless the user explicitly requests and confirms the deletion of this partition. This minimizes accidental erasures.[3] Furthermore, GPT-aware OSes may check the protective MBR and if the enclosed partition type is not of type `EEh` or if there are multiple partitions defined on the target device, the OS may refuse to manipulate the partition table.[6]

If the actual size of the disk exceeds the maximum partition size representable using the legacy 32-bit LBA entries in the MBR partition table, the recorded size of this partition is clipped at the maximum, thereby ignoring the rest of the disk. This amounts to a maximum reported size of 2 TiB, assuming a disk with 512

bytes per sector (see 512e). It would result in 16 TiB with 4 KiB sectors (4Kn), but since many older operating systems and tools are hard coded for a sector size of 512 bytes or are limited to 32-bit calculations, exceeding the 2 TiB limit could cause compatibility problems.[3]

Hybrid MBR (LBA 0 + GPT)

In operating systems that support GPT-based boot through BIOS services rather than EFI, the first sector may also still be used to store the first stage of the bootloader code, but modified to recognize GPT partitions. The bootloader in the MBR must not assume a sector size of 512 bytes.[3]

Partition table header (LBA 1)

GPT header format

Offset	Length	Contents
0 (0x00)	8 bytes	Signature ("EFI PART", `45h 46h 49h 20h 50h 41h 52h 54h` or `0x5452415020494645ULL`[a] on little-endian machines)
8 (0x08)	4 bytes	Revision (for GPT version 1.0 (through at least UEFI version 2.7 (May 2017)), the value is `00h 00h 01h 00h`)
12 (0x0C)	4 bytes	Header size in little endian (in bytes, usually `5Ch 00h 00h 00h` or 92 bytes)
16 (0x10)	4 bytes	CRC32 of header (offset +0 up to header size) in little endian, with this field zeroed during calculation
20 (0x14)	4 bytes	Reserved; must be zero
24 (0x18)	8 bytes	Current LBA (location of this header copy)
32 (0x20)	8 bytes	Backup LBA (location of the other header copy)
40 (0x28)	8 bytes	First usable LBA for partitions (primary partition table last LBA + 1)
48 (0x30)	8 bytes	Last usable LBA (secondary partition table first LBA − 1)
56 (0x38)	16 bytes	Disk GUID in mixed endian[6]
72 (0x48)	8 bytes	Starting LBA of array of partition entries (always 2 in primary copy)
80 (0x50)	4 bytes	Number of partition entries in array
84 (0x54)	4 bytes	Size of a single partition entry (usually `80h` or 128)
88 (0x58)	4 bytes	CRC32 of partition entries array in little endian
92 (0x5C)	*	Reserved; must be zeroes for the rest of the block (420 bytes for a sector size of 512 bytes; but can be more with larger sector sizes)

The partition table header defines the usable blocks on the disk. It also defines the number and size of the partition entries that make up the partition table.

Partition entries (LBA 2–33)

GUID partition entry format

Offset	Length	Contents
0 (0x00)	16 bytes	Partition type GUID (mixed endian[6])
16 (0x10)	16 bytes	Unique partition GUID (mixed endian)
32 (0x20)	8 bytes	First LBA (little endian)
40 (0x28)	8 bytes	Last LBA (inclusive, usually odd)
48 (0x30)	8 bytes	Attribute flags (e.g. bit 60 denotes read-only)
56 (0x38)	72 bytes	Partition name (36 UTF-16LE code units)

After the header, the Partition Entry Array describes partitions, using a minimum size of 128 bytes for each entry block.[7] The starting location of the array on disk, and the size of each entry, are given in the GPT header. The first 16 bytes of each entry designate the partition type's globally unique identifier (GUID). For example, the GUID for an EFI system partition is `C12A7328-F81F-11D2-BA4B-00A0C93EC93B`. The second 16 bytes are a GUID unique to the partition. Then follow the starting and ending 64 bit LBAs, partition attributes, and the 36 character (max.) Unicode partition name. As is the nature and purpose of GUIDs and as per RFC4122[8], no central registry is needed to ensure the uniqueness of the GUID partition type designators.

The 64-bit partition table attributes are shared between 48-bit common attributes for all partition types, and 16-bit type-specific attributes:

Partition attributes

Bit	Content
0	Platform required (required by the computer to function properly, OEM partition for example, disk partitioning utilities must preserve the partition as is)
1	EFI firmware should ignore the content of the partition and not try to read from it
2	Legacy BIOS bootable (equivalent to *active flag* (typically bit 7 set) at offset `+0h` in partition entries of the MBR partition table)[9]
3–47	Reserved for future use
48–63	Defined and used by the individual partition type

Microsoft defines the type-specific attributes for basic data partition as:[10][11]

Basic data partition attributes

Bit	Content
60	Read-only
61	Shadow copy (of another partition)
62	Hidden
63	No drive letter (i.e. do not automount)

Google defines the type-specific attributes for Chrome OS kernel as:[12]

Chrome OS kernel partition attributes

Bit	Content
56	Successful boot flag
55–52	Tries remaining
51–48	Priority (15: highest, 1: lowest, 0: not bootable)

Operating-system support

UNIX and Unix-like systems

Details of GPT support on UNIX and Unix-like operating systems

OS family	Version or edition	Platform	Read and write support	Boot support	Note
FreeBSD	Since 7.0	IA-32, x86-64, ARM	Yes	Yes	In a hybrid configuration, both GPT and MBR partition identifiers may be used.
Linux	Most of the x86 Linux distributions Fedora 8+ and Ubuntu 8.04+[13]	IA-32, x86-64	Yes	Yes	New tools such as gdisk, GNU Parted,[14][15] util-linux v2.23+ fdisk,[16][17] SYSLINUX, GRUB 0.96 + patches and GRUB 2 have been GPT-enabled.
macOS	Since 10.4.0 (some features since 10.4.6)[18]	IA-32, x86-64, PowerPC	Yes	Yes	Only Intel Macintosh computers can boot from GPT.
MidnightBSD	Since 0.4-CURRENT	IA-32, x86-64	Yes	Requires BIOS	In a hybrid configuration, both GPT and MBR partition identifiers may be used.
NetBSD	Since 6.0[19]	x86[20], x86-64[21]	Yes	In progress	Some users describe their installation procedure with EFI[22] but the official announcement of the support has not yet been made. A semi-manual UEFI installation works on NetBSD 8.0 (-current).[23]
OpenBSD	Since 5.9	x86_64	Yes	Requires UEFI	[24]
Solaris	Since Solaris 10	IA-32, x86-64, SPARC	Yes	Yes	[25]
HP-UX	Since HP-UX 11.20	IA-64	Yes	Yes	[26]

Windows: 32-bit versions

Windows 7 and earlier do not support UEFI on 32-bit platforms, and therefore do not allow booting from GPT partitions.[27]

Details of GPT support on 32-bit editions of Microsoft Windows[27]

OS version	Release date	Platform	Read or write support	Boot support	Note
Windows XP	2001-10-25	IA-32	No	No	
Windows Server 2003	2003-04-24	IA-32	No	No	
Windows Server 2003 SP1	2005-03-30	IA-32	Yes	No	MBR takes precedence in hybrid configuration.
Windows Vista	2006-07-22	IA-32	Yes	No	MBR takes precedence in hybrid configuration.
Windows Server 2008	2008-02-27	IA-32	Yes	No	MBR takes precedence in hybrid configuration.
Windows 7	2009-10-22	IA-32	Yes	No	MBR takes precedence in hybrid configuration.
Windows 8	2012-08-01	IA-32	Yes	Requires UEFI[28]	MBR takes precedence in hybrid configuration.
Windows 8.1	2013-08-27	IA-32	Yes	Requires UEFI[28]	MBR takes precedence in hybrid configuration.
Windows 10	2015-07-29	IA-32	Yes	Requires UEFI[28]	MBR takes precedence in hybrid configuration.

Windows: 64-bit versions

Details of GPT support on 64-bit editions of Microsoft Windows[27]

OS version	Release date	Platform	Read and write support	Boot support	Note
Windows XP Professional x64 Edition Windows Server 2003	2005-04-25[29]	x64	Yes	No	MBR takes precedence in hybrid configuration.
Windows Server 2003	2005-04-25	IA-64	Yes	Yes	MBR takes precedence in hybrid configuration.
Windows Vista	2006-07-22	x64	Yes	Requires UEFI[b]	MBR takes precedence in hybrid configuration.
Windows Server 2008	2008-02-27	x64	Yes	Requires UEFI	MBR takes precedence in hybrid configuration.
Windows Server 2008	2008-02-27	IA-64	Yes	Yes	MBR takes precedence in hybrid configuration.
Windows 7	2009-10-22	x64	Yes	Requires UEFI[c]	MBR takes precedence in hybrid configuration.
Windows Server 2008 R2	2009-10-22	IA-64	Yes	Yes	MBR takes precedence in hybrid configuration.
Windows 8 Windows Server 2012	2012-08-01	x64	Yes	Requires UEFI[28]	MBR takes precedence in hybrid configuration.
Windows 8.1	2013-08-27	x64	Yes	Requires UEFI[28]	MBR takes precedence in hybrid configuration.
Windows 10	2015-07-29	x64	Yes	Requires UEFI[28]	MBR takes precedence in hybrid configuration.
Windows Server 2016	2016-10-12	x64	Yes	Requires UEFI[28]	MBR takes precedence in hybrid configuration.

Partition type GUIDs

Each partition has a "partition type GUID" that identifies the type of the partition and therefore partitions of the same type will all have the same "partition type GUID". Each partition also has a "partition unique GUID" as a separate entry, which as the name implies is a unique id for each partition.

Operating system	Partition type	Globally unique identifier (GUID)[d]
N/A	Unused entry	00000000-0000-0000-0000-000000000000
	MBR partition scheme	024DEE41-33E7-11D3-9D69-0008C781F39F
	EFI System partition	C12A7328-F81F-11D2-BA4B-00A0C93EC93B
	BIOS boot partition[e]	21686148-6449-6E6F-744E-656564454649
	Intel Fast Flash (iFFS) partition (for Intel Rapid Start technology)[30][31]	D3BFE2DE-3DAF-11DF-BA40-E3A556D89593
	Sony boot partition[f]	F4019732-066E-4E12-8273-346C5641494F
	Lenovo boot partition[f]	BFBFAFE7-A34F-448A-9A5B-6213EB736C22
Windows	Microsoft Reserved Partition (MSR)	E3C9E316-0B5C-4DB8-817D-F92DF00215AE
	Basic data partition[g]	EBD0A0A2-B9E5-4433-87C0-68B6B72699C7
	Logical Disk Manager (LDM) metadata partition	5808C8AA-7E8F-42E0-85D2-E1E90434CFB3
	Logical Disk Manager data partition	AF9B60A0-1431-4F62-BC68-3311714A69AD
	Windows Recovery Environment	DE94BBA4-06D1-4D40-A16A-BFD50179D6AC
	IBM General Parallel File System (GPFS) partition	37AFFC90-EF7D-4E96-91C3-2D7AE055B174
	Storage Spaces partition	E75CAF8F-F680-4CEE-AFA3-B001E56EFC2D
HP-UX	Data partition	75894C1E-3AEB-11D3-B7C1-7B03A0000000
	Service partition	E2A1E728-32E3-11D6-A682-7B03A0000000
Linux	Linux filesystem data[g]	0FC63DAF-8483-4772-8E79-3D69D8477DE4
	RAID partition	A19D880F-05FC-4D3B-A006-743F0F84911E
	Root partition (x86)[34][35]	44479540-F297-41B2-9AF7-D131D5F0458A
	Root partition (x86-64)[34][35]	4F68BCE3-E8CD-4DB1-96E7-FBCAF984B709
	Root partition (32-bit ARM)[34][35]	69DAD710-2CE4-4E3C-B16C-21A1D49ABED3
	Root partition (64-bit ARM/AArch64)[34][35]	B921B045-1DF0-41C3-AF44-4C6F280D3FAE

	`/boot` partition[34][35]	`BC13C2FF-59E6-4262-A352-B275FD6F7172`
	Swap partition[34][35]	`0657FD6D-A4AB-43C4-84E5-0933C84B4F4F`
	Logical Volume Manager (LVM) partition	`E6D6D379-F507-44C2-A23C-238F2A3DF928`
	`/home` partition[34][35]	`933AC7E1-2EB4-4F13-B844-0E14E2AEF915`
	`/srv` (server data) partition[34][35]	`3B8F8425-20E0-4F3B-907F-1A25A76F98E8`
	Plain dm-crypt partition[36][37][38]	`7FFEC5C9-2D00-49B7-8941-3EA10A5586B7`
	LUKS partition[36][37][38][39]	`CA7D7CCB-63ED-4C53-861C-1742536059CC`
	Reserved	`8DA63339-0007-60C0-C436-083AC8230908`
FreeBSD	Boot partition	`83BD6B9D-7F41-11DC-BE0B-001560B84F0F`
	Data partition	`516E7CB4-6ECF-11D6-8FF8-00022D09712B`
	Swap partition	`516E7CB5-6ECF-11D6-8FF8-00022D09712B`
	Unix File System (UFS) partition	`516E7CB6-6ECF-11D6-8FF8-00022D09712B`
	Vinum volume manager partition	`516E7CB8-6ECF-11D6-8FF8-00022D09712B`
	ZFS partition	`516E7CBA-6ECF-11D6-8FF8-00022D09712B`
macOS Darwin	Hierarchical File System Plus (HFS+) partition	`48465300-0000-11AA-AA11-00306543ECAC`
	Apple APFS container APFS FileVault volume container	`7C3457EF-0000-11AA-AA11-00306543ECAC`
	Apple UFS container	`55465300-0000-11AA-AA11-00306543ECAC`
	ZFS[h]	`6A898CC3-1DD2-11B2-99A6-080020736631`
	Apple RAID partition	`52414944-0000-11AA-AA11-00306543ECAC`
	Apple RAID partition, offline	`52414944-5F4F-11AA-AA11-00306543ECAC`
	Apple Boot partition (Recovery HD)	`426F6F74-0000-11AA-AA11-00306543ECAC`
	Apple Label	`4C616265-6C00-11AA-AA11-00306543ECAC`
	Apple TV Recovery partition	`5265636F-7665-11AA-AA11-00306543ECAC`

	Apple Core Storage Container HFS+ FileVault volume container	53746F72-6167-11AA-AA11-00306543ECAC
	SoftRAID_Status	B6FA30DA-92D2-4A9A-96F1-871EC6486200
	SoftRAID_Scratch	2E313465-19B9-463F-8126-8A7993773801
	SoftRAID_Volume	FA709C7E-65B1-4593-BFD5-E71D61DE9B02
	SoftRAID_Cache	BBBA6DF5-F46F-4A89-8F59-8765B2727503
Solaris illumos	Boot partition	6A82CB45-1DD2-11B2-99A6-080020736631
	Root partition	6A85CF4D-1DD2-11B2-99A6-080020736631
	Swap partition	6A87C46F-1DD2-11B2-99A6-080020736631
	Backup partition	6A8B642B-1DD2-11B2-99A6-080020736631
	`/usr` partition[h]	6A898CC3-1DD2-11B2-99A6-080020736631
	`/var` partition	6A8EF2E9-1DD2-11B2-99A6-080020736631
	`/home` partition	6A90BA39-1DD2-11B2-99A6-080020736631
	Alternate sector	6A9283A5-1DD2-11B2-99A6-080020736631
	Reserved partition	6A945A3B-1DD2-11B2-99A6-080020736631
		6A9630D1-1DD2-11B2-99A6-080020736631
		6A980767-1DD2-11B2-99A6-080020736631
		6A96237F-1DD2-11B2-99A6-080020736631
		6A8D2AC7-1DD2-11B2-99A6-080020736631
NetBSD[40][i]	Swap partition	49F48D32-B10E-11DC-B99B-0019D1879648
	FFS partition	49F48D5A-B10E-11DC-B99B-0019D1879648
	LFS partition	49F48D82-B10E-11DC-B99B-0019D1879648
	RAID partition	49F48DAA-B10E-11DC-B99B-0019D1879648
	Concatenated partition	2DB519C4-B10F-11DC-B99B-0019D1879648

	Encrypted partition	2DB519EC-B10F-11DC-B99B-0019D1879648
Chrome OS[41]	Chrome OS kernel	FE3A2A5D-4F32-41A7-B725-ACCC3285A309
	Chrome OS rootfs	3CB8E202-3B7E-47DD-8A3C-7FF2A13CFCEC
	Chrome OS future use	2E0A753D-9E48-43B0-8337-B15192CB1B5E
Container Linux by CoreOS[42]	/usr partition (coreos-usr)	5DFBF5F4-2848-4BAC-AA5E-0D9A20B745A6
	Resizable rootfs (coreos-resize)	3884DD41-8582-4404-B9A8-E9B84F2DF50E
	OEM customizations (coreos-reserved)	C95DC21A-DF0E-4340-8D7B-26CBFA9A03E0
	Root filesystem on RAID (coreos-root-raid)	BE9067B9-EA49-4F15-B4F6-F36F8C9E1818
Haiku[43]	Haiku BFS	42465331-3BA3-10F1-802A-4861696B7521
MidnightBSD[44][i]	Boot partition	85D5E45E-237C-11E1-B4B3-E89A8F7FC3A7
	Data partition	85D5E45A-237C-11E1-B4B3-E89A8F7FC3A7
	Swap partition	85D5E45B-237C-11E1-B4B3-E89A8F7FC3A7
	Unix File System (UFS) partition	0394EF8B-237E-11E1-B4B3-E89A8F7FC3A7
	Vinum volume manager partition	85D5E45C-237C-11E1-B4B3-E89A8F7FC3A7
	ZFS partition	85D5E45D-237C-11E1-B4B3-E89A8F7FC3A7
Ceph[j]	Journal	45B0969E-9B03-4F30-B4C6-B4B80CEFF106
	dm-crypt journal	45B0969E-9B03-4F30-B4C6-5EC00CEFF106
	OSD	4FBD7E29-9D25-41B8-AFD0-062C0CEFF05D
	dm-crypt OSD	4FBD7E29-9D25-41B8-AFD0-5EC00CEFF05D
	Disk in creation	89C57F98-2FE5-4DC0-89C1-F3AD0CEFF2BE
	dm-crypt disk in creation	89C57F98-2FE5-4DC0-89C1-5EC00CEFF2BE
	Block	CAFECAFE-9B03-4F30-B4C6-B4B80CEFF106
	Block DB	30CD0809-C2B2-499C-8879-2D6B78529876

	Block write-ahead log	5CE17FCE-4087-4169-B7FF-056CC58473F9
	Lockbox for dm-crypt keys	FB3AABF9-D25F-47CC-BF5E-721D1816496B
	Multipath OSD	4FBD7E29-8AE0-4982-BF9D-5A8D867AF560
	Multipath journal	45B0969E-8AE0-4982-BF9D-5A8D867AF560
	Multipath block	CAFECAFE-8AE0-4982-BF9D-5A8D867AF560
	Multipath block	7F4A666A-16F3-47A2-8445-152EF4D03F6C
	Multipath block DB	EC6D6385-E346-45DC-BE91-DA2A7C8B3261
	Multipath block write-ahead log	01B41E1B-002A-453C-9F17-88793989FF8F
	dm-crypt block	CAFECAFE-9B03-4F30-B4C6-5EC00CEFF106
	dm-crypt block DB	93B0052D-02D9-4D8A-A43B-33A3EE4DFBC3
	dm-crypt block write-ahead log	306E8683-4FE2-4330-B7C0-00A917C16966
	dm-crypt LUKS journal	45B0969E-9B03-4F30-B4C6-35865CEFF106
	dm-crypt LUKS block	CAFECAFE-9B03-4F30-B4C6-35865CEFF106
	dm-crypt LUKS block DB	166418DA-C469-4022-ADF4-B30AFD37F176
	dm-crypt LUKS block write-ahead log	86A32090-3647-40B9-BBBD-38D8C573AA86
	dm-crypt LUKS OSD	4FBD7E29-9D25-41B8-AFD0-35865CEFF05D
OpenBSD	Data partition	824CC7A0-36A8-11E3-890A-952519AD3F61
QNX	Power-safe (QNX6) file system[46]	CEF5A9AD-73BC-4601-89F3-CDEEEEE321A1
Plan 9	Plan 9 partition	C91818F9-8025-47AF-89D2-F030D7000C2C
VMware ESX	vmkcore (coredump partition)	9D275380-40AD-11DB-BF97-000C2911D1B8
	VMFS filesystem partition	AA31E02A-400F-11DB-9590-000C2911D1B8
	VMware Reserved	9198EFFC-31C0-11DB-8F78-000C2911D1B8
Android-IA[47][48][49][50]	Bootloader	2568845D-2332-4675-BC39-8FA5A4748D15

	Bootloader2	114EAFFE-1552-4022-B26E-9B053604CF84
	Boot	49A4D17F-93A3-45C1-A0DE-F50B2EBE2599
	Recovery	4177C722-9E92-4AAB-8644-43502BFD5506
	Misc	EF32A33B-A409-486C-9141-9FFB711F6266
	Metadata	20AC26BE-20B7-11E3-84C5-6CFDB94711E9
	System	38F428E6-D326-425D-9140-6E0EA133647C
	Cache	A893EF21-E428-470A-9E55-0668FD91A2D9
	Data	DC76DDA9-5AC1-491C-AF42-A82591580C0D
	Persistent	EBC597D0-2053-4B15-8B64-E0AAC75F4DB1
	Vendor	C5A0AEEC-13EA-11E5-A1B1-001E67CA0C3C
	Config	BD59408B-4514-490D-BF12-9878D963F378
	Factory	8F68CC74-C5E5-48DA-BE91-A0C8C15E9C80
	Factory (alt)[51]	9FDAA6EF-4B3F-40D2-BA8D-BFF16BFB887B
	Fastboot / Tertiary[52][53]	767941D0-2085-11E3-AD3B-6CFDB94711E9
	OEM	AC6D7924-EB71-4DF8-B48D-E267B27148FF
Android 6.0+ ARM	Android Meta	19A710A2-B3CA-11E4-B026-10604B889DCF
	Android EXT	193D1EA4-B3CA-11E4-B075-10604B889DCF
Open Network Install Environment (ONIE)	Boot	7412F7D5-A156-4B13-81DC-867174929325
	Config	D4E6E2CD-4469-46F3-B5CB-1BFF57AFC149
PowerPC	PReP boot	9E1A2D38-C612-4316-AA26-8B49521E5A8B
freedesktop.org OSes (Linux, etc.)	Shared boot loader configuration[54]	BC13C2FF-59E6-4262-A352-B275FD6F7172
Atari TOS	Basic data partition (GEM, BGM, F32)	734E5AFE-F61A-11E6-BC64-92361F002671
VeraCrypt	Encrypted data partition	8C8F8EFF-AC95-4770-814A-21994F2DBC8F

See also

- Advanced Active Partition (AAP)
- Apple Partition Map (APM)
- Boot Engineering Extension Record (BEER)
- BSD disklabel
- Device Configuration Overlay (DCO)
- Extended Boot Record (EBR)
- Host Protected Area (HPA)
- Partition alignment
- Rigid Disk Block (RDB)
- Volume Table of Contents (VTOC)

Notes

a. Adding `ULL` suffix to an integer constant makes it of type `unsigned long long int`.

b. Only if using its service pack 1 or 2

c. In a multi-disk setup, non-UEFI bootloader (boot drive) requires MBR-based partitioning, while a system drive can use GUID partitioning.

d. The GUIDs in this table are written as per RFC 4122, i.e. big-endian byte order, recognizable by the position of the version bits. For example, the GUID for an EFI System partition (`C12A7328-F81F-11D2-BA4B-00A0C93EC93B`), when serialized in GPT data structures (little-endian), corresponds to the hex sequence `28 73 2A C1 1F F8 D2 11 BA 4B 00 A0 C9 3E C9 3B`. The first three blocks are byte-swapped to little-endian, the last is a byte array. See details in TN2166[6]

e. The formation of this GUID does not follow the GUID definition; it is formed by using the ASCII codes for the string "`Hah!IdontNeedEFI`". Such formation of "GUID" value breaks down the guaranteed uniqueness of GUID.

f. Some computer manufacturers have their own GUIDs for partitions that are analogous to the EFI System Partition, but that hold boot loaders to launch manufacturer-specific recovery tools.[32]

g. Previously, Linux used the same GUID for the data partitions as Windows (Basic data partition: `EBD0A0A2-B9E5-4433-87C0-68B6B72699C7`). Linux never had a separate unique partition type GUID defined for its data partitions. This created problems when dual-booting Linux and Windows in UEFI-GPT setup. The new GUID (Linux filesystem data: `0FC63DAF-8483-4772-8E79-3D69D8477DE4`) was defined jointly by GPT fdisk and GNU Parted developers.[33] It is identified as type code `0x8300` in GPT fdisk. (See definitions in gdisk's parttypes.cc (http://gptfdisk.git.sourceforge.net/git/gitweb.cgi?p%3Dgptfdisk/gptfdisk;a%3Dblob_plain;f%3Dparttypes.cc;hb%3DHEAD))

h. The GUID for `/usr` on Solaris is used as a generic GUID for ZFS by macOS.

i. NetBSD and MidnightBSD had used the FreeBSD GUIDs before their unique GUIDs were created.

j. The Ceph filesystem uses GUIDs to mark the state of preparation a disk is in.[45]

References

1. "FAQ: Drive Partition Limits" (http://www.uefi.org/sites/default/files/resources/UEFI_Drive_Partition_Limits_Fact_Sheet.pdf) (PDF). UEFI Forum. Retrieved 2013-11-04.

2. Nikkel, Bruce J. (September 2009). "Forensic analysis of GPT disks and GUID partition tables". *Digital Investigation*. **6** (1–2): 39–47. doi:10.1016/j.diin.2009.07.001 (https://doi.org/10.1016%2Fj.diin.2009.07.001). "The current popular BIOS and MBR partitioning scheme was originally developed in the early 1980s for the IBM Personal Computer using IBM PC DOS or MS-DOS. The Basic Input/Output System (BIOS) provides an interface to the hardware and initiates the boot process (IBM, 1983). The MBR, located in sector zero, contains the initial boot code and a four entry partition table (Microsoft, 1983). Intended to solve booting and partitioning limitations with newer hardware, a replacement for both the BIOS and the MBR partition table was developed by Intel in the late 1990s (Intel, 2000). This is now called the Unified EFI (UEFI, 2008 UEFI Forum. Unified extensible firmware interface specification version 2.2 2008.UEFI, 2008) specification, and managed by the UEFI Forum (UEFI, 2009). A subset of this specification includes GPT, intended to replace the DOS/MBR partition tables."
3. Smith, Roderick W. (2012-07-03). "Make the Most of Large Drives with GPT and Linux" (http://www.ibm.com/developerworks/library/l-gpt/). IBM. Retrieved 2013-05-29.
4. "UEFI specification" (http://www.uefi.org/specifications). *UEFI.org*.
5. "Western Digital's Advanced Format: The 4K Sector Transition Begins" (http://anandtech.com/storage/showdoc.aspx?i=3691). *Anandtech.com*. Anandtech.
6. "Technical Note TN2166: Secrets of the GPT" (https://developer.apple.com/library/mac/technotes/tn2166/_index.html#//apple_ref/doc/uid/DTS10003927-CH1-SUBSECTION11). *Developer.Apple.com*. Apple. 2006-11-06. Retrieved 2014-04-16.
7. The GPT header contains a field that specifies the size of a partition table entry. The minimum required is 128 bytes, but implementations must allow for other values. See "Mac Developer Library" (https://developer.apple.com/mac/library/technotes/tn2006/tn2166.html#SECGPTOVERVIEW). *Developer.Apple.com*. Apple. Retrieved 2014-07-13.
8. "RFC 4122" (https://www.ietf.org/rfc/rfc4122.txt). Retrieved 2018-10-23.
9. "e09127r3 EDD-4 Hybrid MBR Boot Code Annex" (http://t13.org/documents/UploadedDocuments/docs2010/e09127r3-EDD-4_Hybrid_MBR_boot_code_annex.pdf) (PDF). *T13.org*.
10. https://technet.microsoft.com/en-us/library/cc753455.aspx#Anchor_1
11. https://msdn.microsoft.com/en-us/library/aa381635.aspx
12. "Disk Format" (https://www.chromium.org/chromium-os/chromiumos-design-docs/disk-format#TOC-Selecting-the-kernel). *Chromium.org*. Retrieved 2017-10-04.
13. "Ubuntu on MacBook" (https://help.ubuntu.com/community/MacBook). *Community Documentation*. Ubuntu.
14. "GNU Parted FAQ" (https://www.gnu.org/software/parted/faq.shtml#features).
15. "mklabel" (https://www.gnu.org/software/parted/manual/html_node/mklabel.html). *Parted Manual*. GNU.
16. "fdisk: add GPT support" (https://git.kernel.org/cgit/utils/util-linux/util-linux.git/commit/?id=766d5156c43b784700d28d1c1141008b2bf35ed7). kernel.org. 2013-09-27. Retrieved 2013-10-18.
17. Bueso, Davidlohr (2013-09-28). "fdisk updates and GPT support" (http://blog.stgolabs.net/2012/09/fdisk-updates-and-gpt-support.html). Retrieved 2013-10-18.
18. "Myths and Facts About Intel Macs" (http://refit.sourceforge.net/myths/). *rEFIt*. Source forge.
19. "Significant changes from NetBSD 5.0 to 6.0" (http://www.netbsd.org/changes/changes-6.0.html)..
20. "Significant changes from NetBSD 5.0 to 6.0 (NetBSD/i386)" (http://www.netbsd.org/changes/changes-6.0.html#port-i386)..
21. "Significant changes from NetBSD 5.0 to 6.0 (NetBSD/amd64)" (http://www.netbsd.org/changes/changes-6.0.html#port-amd64)..
22. "How to install NetBSD/amd64 current on MacBook Air 11 inch" (https://wiki.netbsd.org/users/ryoon/how_to_install_netbsd_amd64_to_macbook_air_11_inch/)..

23. "NetBSD Wiki - Installing NetBSD 8.0 on a x86 system with UEFI" (http://wiki.netbsd.org/Installation_on_UEFI_systems/).
24. "OpenBSD 5.9" (https://www.openbsd.org/59.html)..
25. "Booting from a ZFS Root File System" (http://docs.oracle.com/cd/E19963-01/html/821-1448/ggpco.html). Oracle.
26. "idisk(1M)" (http://h20000.www2.hp.com/bc/docs/support/SupportManual/c02256075/c02256075.pdf) (PDF). Hewlett-Packard.
27. "Windows and GPT FAQ" (https://msdn.microsoft.com/en-us/library/windows/hardware/dn640535.aspx). Microsoft.
28. Windows 8 32-bit supports booting from UEFI-based PC using GPT-based disks. (https://technet.microsoft.com/en-us/library/hh824898.aspx)
29. Microsoft raises the speed limit with the availability of 64-bit editions of Windows Server 2003 and Windows XP Professional (http://www.microsoft.com/presspass/press/2005/apr05/04-25Winx64LaunchPR.mspx) Archived (https://web.archive.org/web/20101110033856/http://www.microsoft.com/presspass/press/2005/apr05/04-25Winx64LaunchPR.mspx) 2010-11-10 at the Wayback Machine
30. ftp://download.gigabyte.ru/manual/mb_manual_intel-ui_e.pdf
31. "F6F: Funtoo Linux and Intel Rapid Start Technology" (http://blog.adios.tw/2012/10/funtoo-linux-and-intel-rapid-start.html). Blog.adios.tw. 2012-10-30. Retrieved 2014-01-29.
32. GPT fdisk: parttypes.cc, line 198 (http://sourceforge.net/p/gptfdisk/code/ci/master/tree/parttypes.cc)
33. Smith, Rod (23 June 2011). "Need for a unique Linux GPT GUID type code (PATCH included)" (http://lists.gnu.org/archive/html/bug-parted/2011-06/msg00026.html). *bug-parted* (Mailing list). lists.gnu.org. Retrieved 12 April 2016.
34. The Discoverable Partitions Specification (http://www.freedesktop.org/wiki/Specifications/DiscoverablePartitionsSpec/)
35. systemd-gpt-auto-generator(8) (https://www.freedesktop.org/software/systemd/man/systemd-gpt-auto-generator.html)
36. "[dm-crypt] LUKS GPT GUID" (http://www.saout.de/pipermail/dm-crypt/2014-January/003855.html). Saout.de. Retrieved 2014-01-29.
37. "[dm-crypt] LUKS GPT GUID" (http://www.saout.de/pipermail/dm-crypt/2014-January/003859.html). Saout.de. Retrieved 2014-01-29.
38. "pyuefi source code" (https://bitbucket.org/mmabey/pyuefi/src/master/pyuefi/partition_types/uefi.py).
39. "udisks-2.7.4 source code" (https://github.com/storaged-project/udisks/blob/udisks-2.7.4/src/udiskslinuxblock.c#L2363).
40. "CVS log for src/sys/sys/disklabel_gpt.h" (http://cvsweb.netbsd.org/bsdweb.cgi/src/sys/sys/disklabel_gpt.h?only_with_tag=MAIN). Cvsweb.netbsd.org. Retrieved 2014-01-29.
41. "Disk Format - The Chromium Projects" (https://www.chromium.org/chromium-os/chromiumos-design-docs/disk-format). Chromium.org. Retrieved 2014-01-29.
42. "Constants and IDs" (https://coreos.com/os/docs/latest/constants-and-ids.html#gpt-partition-types). CoreOS. Retrieved 2018-07-26.
43. src/add-ons/kernel/partitioning_systems/gpt/gpt_known_guids.h (http://cgit.haiku-os.org/haiku/tree/src/add-ons/kernel/partitioning_systems/gpt/gpt_known_guids.h)
44. http://www.midnightbsd.org/cgi-bin/cvsweb.cgi/src/sys/sys/gpt.h.diff?r1=1.4;r2=1.5 src/sys/sys/gpt.h
45. Script to set up a ceph disk: ceph-disk, lines 76-81 (https://github.com/ceph/ceph/blob/9bcc42a3e6b08521694b5c0228b2c6ed7b3d312e/src/ceph-disk#L76-L81)
46. QNX Power-safe filesystem (http://www.qnx.com/developers/docs/660/index.jsp?topic=%2Fcom.qnx.doc.neutrino.user_guide%2Ftopic%2Ffsystems_QNX6_filesystem.html)

47. "gpt.ini (github.com/android-ia/device-androidia-mixins)" (https://github.com/android-ia/device-androidia-mixins/blob/master/groups/boot-arch/android_ia/gpt.ini).
48. "gpt.ini (github.com/android-ia/device-androidia)" (https://github.com/android-ia/device-androidia/blob/master/androidia_64/gpt.ini).
49. "gpt.ini (github.com/android-ia/vendor_intel_baytrail)" (https://github.com/android-ia/vendor_intel_baytrail/blob/master/minnow_max/gpt.ini).
50. "gpt-sample.ini (github.com/android-ia/platform_bootable_userfastboot)" (https://github.com/android-ia/platform_bootable_userfastboot/blob/master/gpt-sample.ini).
51. "gpt_ini2bin.py (android.googlesource.com/platform/hardware/bsp/intel)" (https://android.googlesource.com/platform/hardware/bsp/intel/+/e2ef91a5723be4c50e0bd93f82c329d262c2f366/soc/common/tools/gpt_ini2bin.py).
52. "gpt.c (github.com/android-ia/platform_bootable_userfastboot)" (https://github.com/android-ia/platform_bootable_userfastboot/blob/master/libgpt/gpt.c).
53. "gpt_ini2bin.py (github.com/android-ia/vendor_intel_common)" (https://github.com/android-ia/vendor_intel_common/blob/master/gpt_bin/gpt_ini2bin.py).
54. "The Boot Loader Specification" (https://freedesktop.org/wiki/Specifications/BootLoaderSpec/). *freedesktop.org*. Retrieved 2017-01-05.

External links

- Microsoft TechNet: Disk Sectors on GPT Disks (archived page) (https://web.archive.org/web/20080321063028/http://technet2.microsoft.com/windowsserver/en/library/bdeda920-1f08-4683-9ffb-7b4b50df0b5a1033.mspx?mfr=true)
- Microsoft TechNet: Troubleshooting Disks and File Systems (https://technet.microsoft.com/en-us/library/bb457122.aspx)
- Microsoft TechNet: Using GPT Drives (https://docs.microsoft.com/en-us/previous-versions/windows/hardware/design/dn653580(v=vs.85))
- Microsoft: FAQs on Using GPT disks in Windows (http://www.microsoft.com/whdc/device/storage/GPT_FAQ.mspx)
- Microsoft Technet: How Basic Disks and Volumes Work (https://technet.microsoft.com/en-us/library/cc739412.aspx) A bit MS-specific but good figures relate GPT to older MBR format and protective-MBR, shows layouts of complete disks, and how to interpret partition-table hexdumps.
- Apple Developer Connection: Secrets of the GPT (https://developer.apple.com/technotes/tn2006/tn2166.html)
- Make the most of large drives with GPT and Linux (http://www.ibm.com/developerworks/linux/library/l-gpt/)
- Convert Windows Vista SP1+ or 7 x86_64 boot from BIOS-MBR mode to UEFI-GPT mode without Reinstall (https://web.archive.org/web/20120426234357/https://gitorious.org/tianocore_uefi_duet_builds/pages/Windows_x64_BIOS_to_UEFI)
- Support for GPT (Partition scheme) and HDD greater than 2.19 TB in Microsoft Windows XP (http://www.ghacks.net/2010/11/04/how-to-use-3tb-hard-drives-on-windows-xp/)
- Setting up a RAID volume in Linux with >2TB disks (https://web.archive.org/web/20111119090413/http://www.technotes.se/?p=1732)