BIOS UEFI MBR and GPT

Contents

- Booting
- PC Firmware
  - BIOS/UEFI
- Disk organisation
  - MBR/GPT
- EFI
  - Accessing

Booting

- Booting is the process of loading an operating system. It’s the process that starts when we turn on the computer (using the power button or by a software command) and ends when the operating system is loaded into the memory.
- When we turn on the computer, there is no program inside the computer’s main memory (RAM), so the CPU looks for another program, in the firmware (EEPROM) that is located on the motherboard and is run by the CPU to start the booting sequence.
- The boot program then reads the Disk to load the boot loader program and execute it.

Booting

- The boot loader program then loads the respective operating system and hands over control to the operating system
- There is no basic difference in this whether the PC is a BIOS system or an UEFI system
- The only difference is in how these two systems do it

PC Firmware

- There currently two main types of firmware in IBM compatible PC's :

  BIOS - Basic Input Output System (1981)

  UEFI - Unified Extensible Firmware Interface (2005/2015)

  see - https://www.intel.com/content/dam/www/public/us/en/zip/efi-1-10-update.zip

- There is however another firmware type, an opensource firmware "Coreboot".

  see https://doc.coreboot.org/

- This presentation will only be about BIOS and UEFI.

PC Firmware

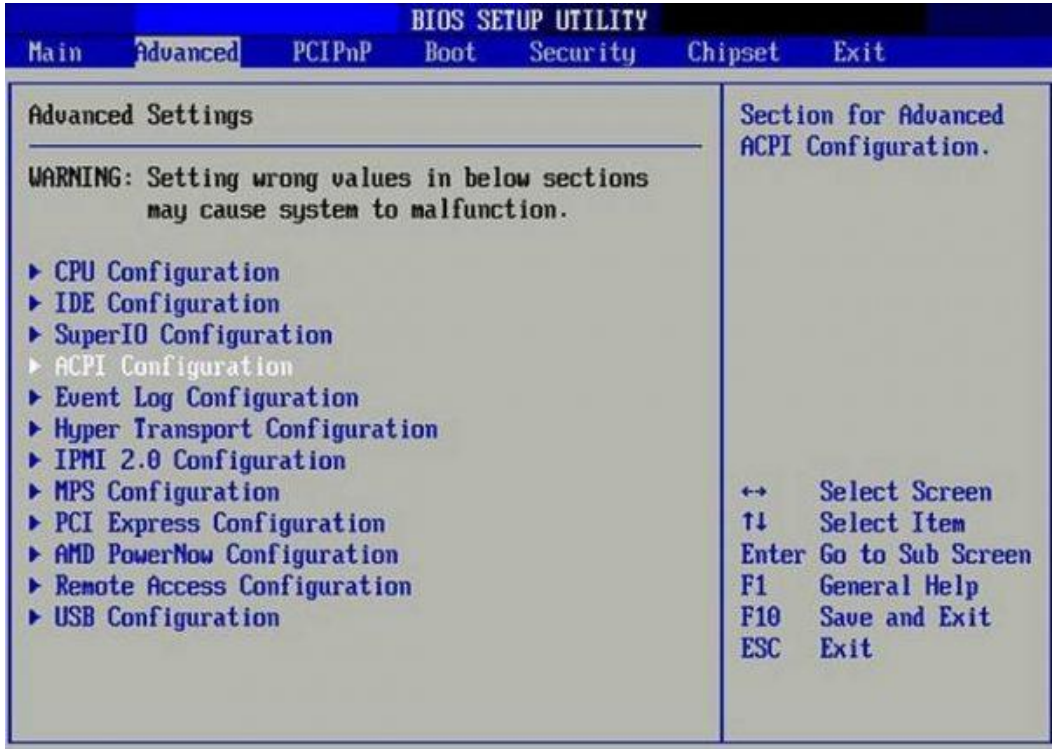
- BIOS and UEFI are two very different systems although they both have basically the same function
- UEFI although being much newer and having a different boot mode it still supports the old BIOS style of booting. This may be called Legacy, BIOS mode, or CSM (Compatibility Support Module) or something similar.

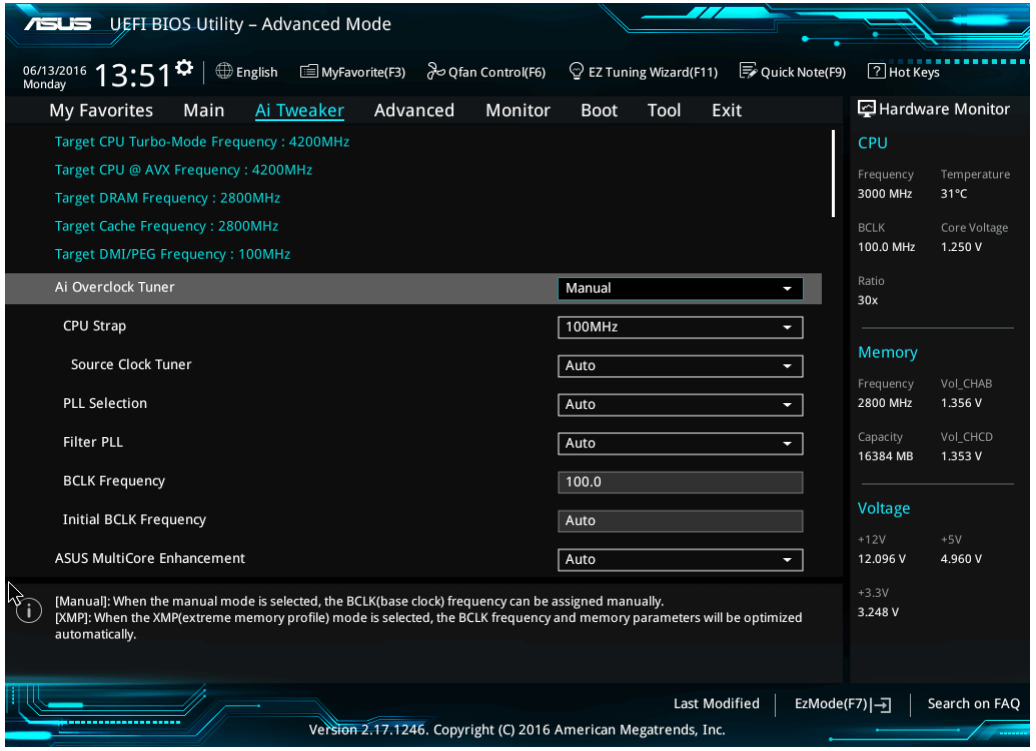
* Mac computers also use UEFI.

PC Firmware

- UEFI offers more features and benefits, such as faster boot times, better security, larger disk support(GPT only), and a graphical user interface. UEFI can often be directly accessed from within the OS, BIOS cannot.
- Some OS's (versions) will only work with UEFI! (w11)
- Others cannot
- Some work with both!
- Legacy BIOS is the old mode that uses a 16-bit code and has a limited number of options, but is still used (OS/2, Windows 7, Win 95, ME..)

BIOS/UEFI


BIOS SETUP UTILITY
Main Advanced PCIPnP Boot Security Chipset Exit
Advanced Settings
WARNING: Setting wrong values in below sections may cause system to malfunction.
CPU Configuration
IDE Configuration
SuperIO Configuration
ACPI Configuration
Event Log Configuration
Hyper Transport Configuration
IPMI 2.0 Configuration
MPS Configuration
PCI Express Configuration
AMD PowerNow Configuration
Remote Access Configuration
USB Configuration
Section for Advanced ACPI Configuration.
↔ Select Screen
↑↓ Select Item
Enter Go to Sub Screen
F1 General Help
F10 Save and Exit
ESC Exit


UEFI BIOS Utility – Advanced Mode
06/13/2016 Monday 13:51
English
MyFavorite(F3)
Qfan Control(F6)
EZ Tuning Wizard(F11)
Quick Note(F9)
Hot Keys
My Favorites Main Ai Tweaker Advanced Monitor Boot Tool Exit
Target CPU Turbo-Mode Frequency : 4200MHz
Target CPU @ AVX Frequency : 4200MHz
Target DRAM Frequency : 2800MHz
Target Cache Frequency : 2800MHz
Target DMI/PEG Frequency : 100MHz
Ai Overclock Tuner Manual
CPU Strap 100MHz
Source Clock Tuner Auto
PLL Selection Auto
Filter PLL Auto
BCLK Frequency 100.0
Initial BCLK Frequency Auto
ASUS MultiCore Enhancement Auto
[Manual]: When the manual mode is selected, the BCLK(base clock) frequency can be assigned manually.
[XMP]: When the XMP(extreme memory profile) mode is selected, the BCLK frequency and memory parameters will be optimized automatically.
Hardware Monitor
CPU
Frequency 3000 MHz
Temperature 31°C
BCLK 100.0 MHz
Core Voltage 1.250 V
Ratio 30x
Memory
Frequency 2800 MHz
Vol_CHAB 1.356 V
Capacity 16384 MB
Vol_CHCD 1.353 V
Voltage
+12V 12.096 V
+5V 4.960 V
+3.3V 3.248 V
Last Modified
EzMode(F7)
Search on FAQ
Version 2.17.1246. Copyright (C) 2016 American Megatrends, Inc.

PC Firmware

- In order to boot, the firmware reads the boot loader from the disk, how depends on the disk layout of which there are two:

  MBR - Master Boot Record
  GPT - The GUID (Globally Unique Identifier) Partition Table

- BIOS can only access MBR but UEFI can access both

MBR – booting an OS

- When the system boots using BIOS (UEFI legacy, or compatibility mode) it is running in 16 bit mode.
- It uses the program in the NVRAM on the motherboard which first performs the POST(Power On Self Test) process.
- Next the BIOS goes through each disk in a predetermined order (boot order), and loads the first sector (first 512 bytes) and puts it into RAM at address 0x7C00, if and only if those 512 bytes end in 0xAA55 aka the boot signature.
- If the signature is not found it tries the next disk, and so on. When found the BIOS transfers control to this first stage bootloader. In other words, opcode located at physical memory address 0x7C00 in DRAM is executed.

MBR- UEFI Misconceptions

- UEFI requires GPT
- UEFI Cannot boot with MBR
- BIOS cannot use GPT

MBR – booting an OS

- Communication between the OS and BIOS is accomplished via interrupts
- For OS/2 Interrupt 10 is VGA hardware access, (including DOS full screen)
- For OS/2 Interrupt 13 is generic disk access (including bootstrap and trap dump)

BIOS Interrupts

Interrupt Vector	Description
05h	Shift-Print screen / BOUND failure.
08h	Real time clock interrupt.
09h	Keyboard interrupt.
10h	Video Services - Video Mode, Cursor Shape/Position, Get Video Mode, Palette Registers (EGA, VGA, SVGA) .....
11h	Returns equipment list
12h	Return conventional memory size
13h	Low Level Disk Services
14h	Serial port services
15h	Miscellaneous system services
16h	Keyboard services
17h	Printer services

Interrupt Vector	Description
18h	Execute Cassette BASIC:
19h	Load the operating system.
1Ah	Real Time Clock Services
1Ah	PCI Services
1Bh	Ctrl-Break handler
1Ch	Timer tick handler
1Dh	Not to be called;
1Eh	Not to be called;
1Fh	Not to be called;
41h	Address pointer: FDPT = Fixed Disk Parameter Table (1st hard drive)
46h	Address pointer: FDPT = Fixed Disk Parameter Table (2nd hard drive)
4Ah	Called by RC for alarm

UEFI – booting an OS

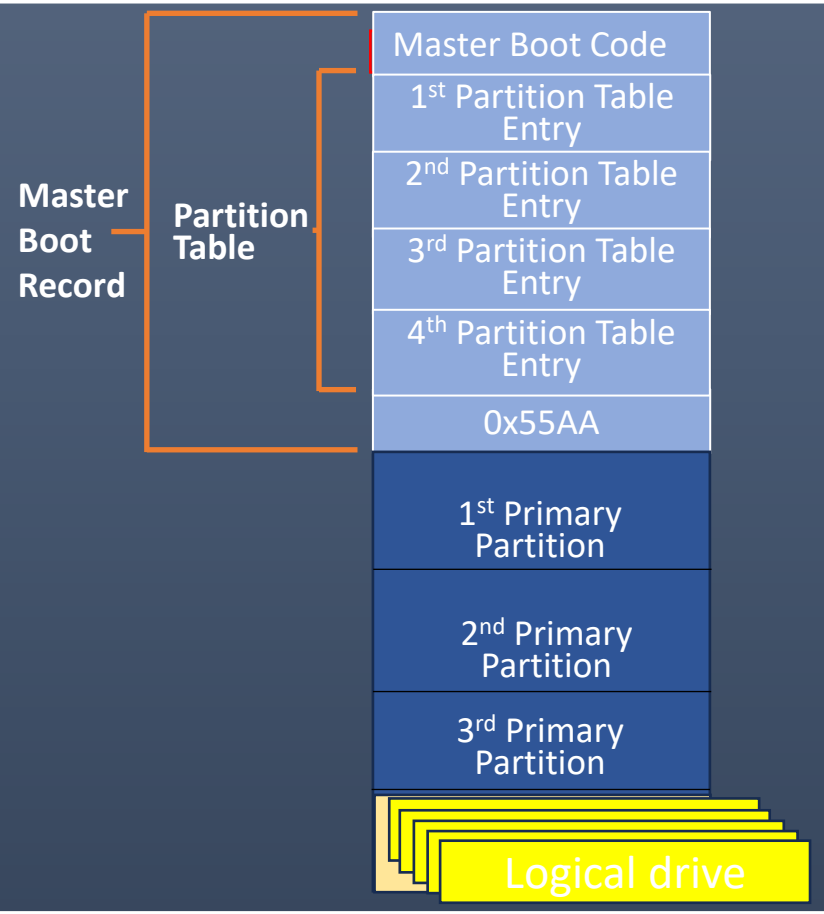
- When the system boots using UEFI it also starts in Real Mode after which the UEFI builds a rudimentary operating system on the platform to enable 64-bit Protected Mode.
- UEFI first scans for available boot entries in the NVRAM, which can include operating systems installed on different disks or partitions.
- It then uses the default entry to select the desired boot from the selected directory from the special GPT EFI partition.
- Communication between the OS and UEFI is accomplished via UEFI services not interrupts!

UEFI – booting ArcaOS

- In the case of ArcaOS/OS2, the boot loader relies heavily on a number of functions supplied by the BIOS (mainly int 10 and int 13) until the necessary device drives have been loaded and are available. Thereafter the BIOS functions are no longer required.
- These functions are not available from an UEFI firmware. This means it is impossible to load OS/2 on a UEFI system. However by adding the missing functionality of the BIOS functions to ArcaOS (5.1) it is now possible to boot from UEFI
- Booting under UEFI requires a EFI System Partition (FAT32)

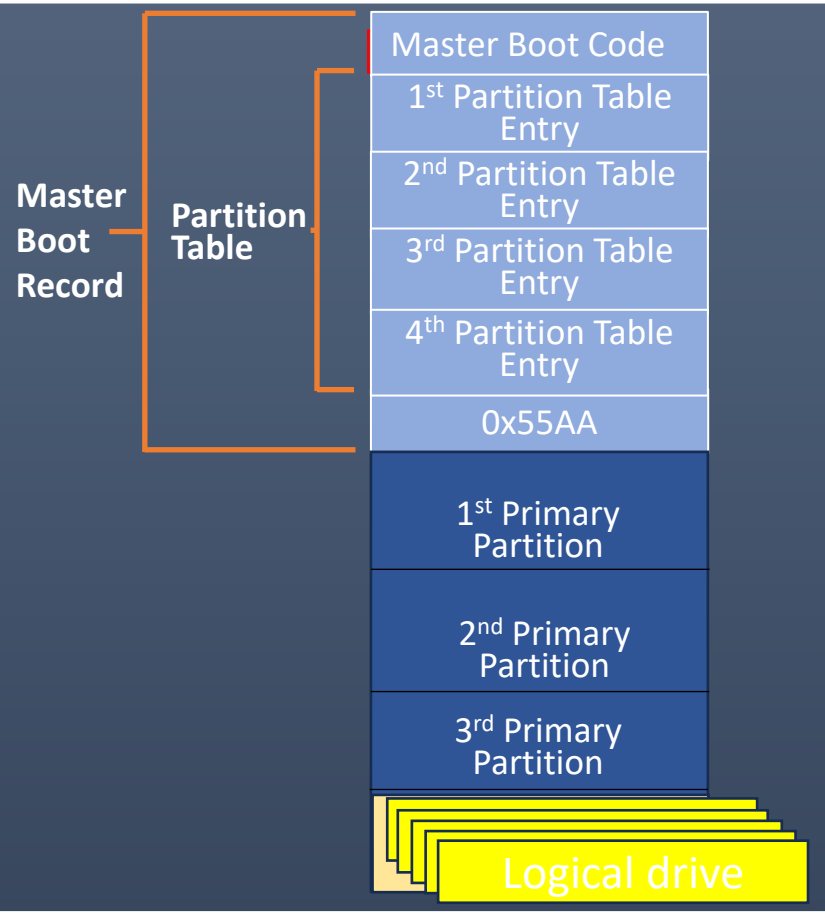
Disk organisation - MBR

- In the very first sectors of an MBR disk is the boot code to boot the system. The BIOS executes this code which starts the specific OS boot procedure
- A MBR Disk has a maximum of 4 primary partitions.
- Often one primary partition called the extended partition, is divided into a number of logical partitions



MBR

- A MBR disk uses 32 bits to describe the starting point and offset and can only access a maximum of 2^{32} bytes, about 2.19TB

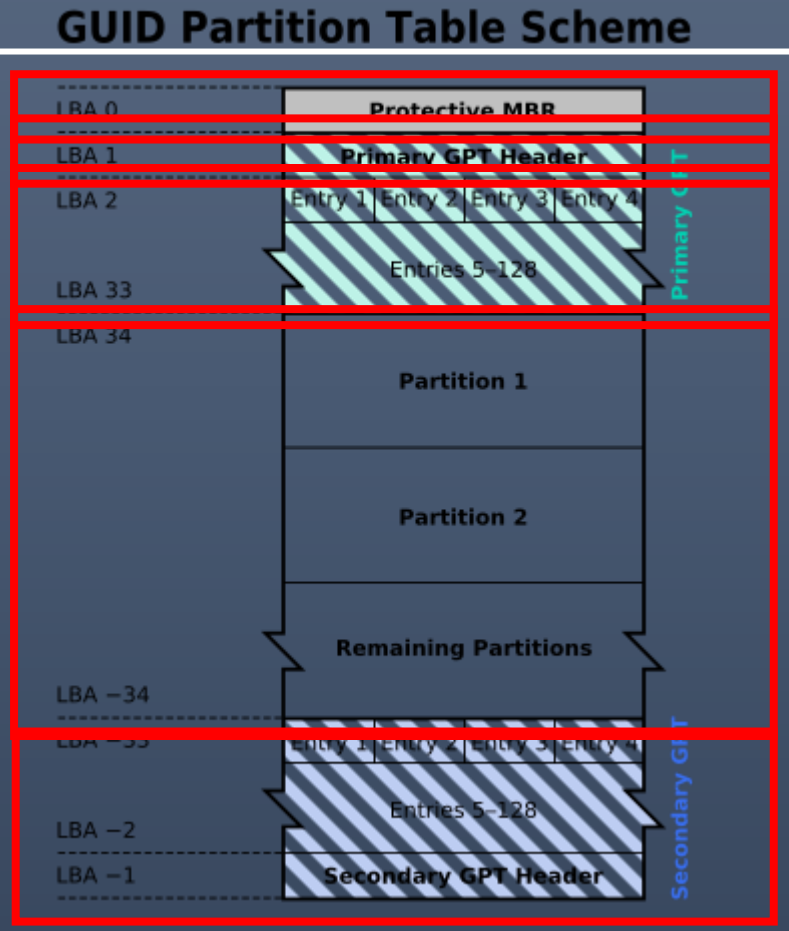


Disk organisation - GPT

A GPT-disk contains:

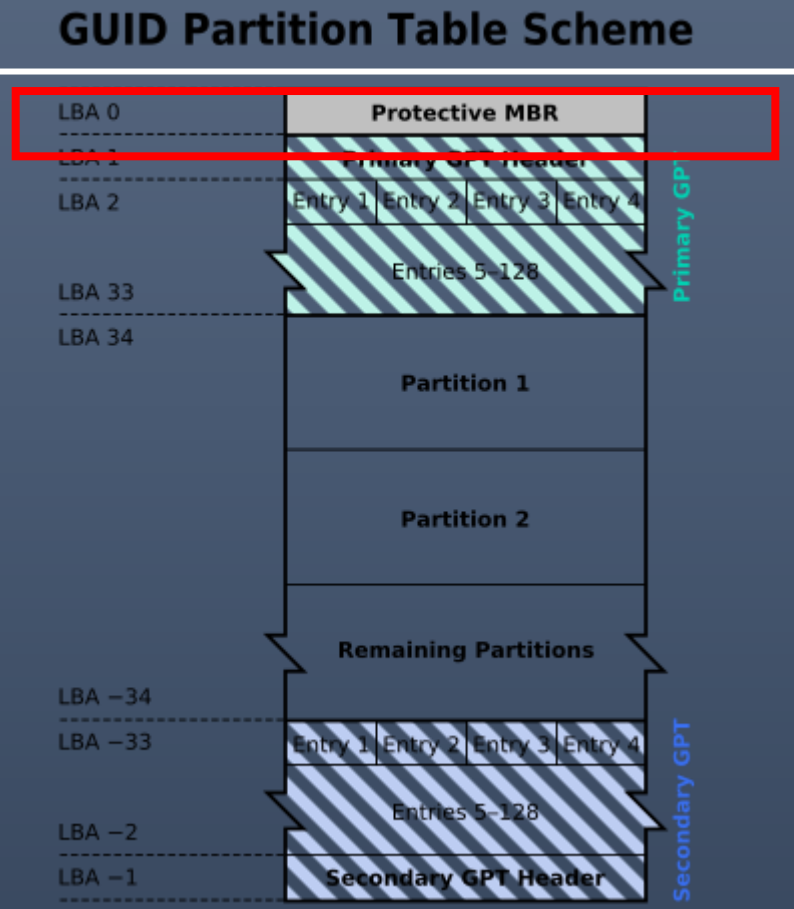
- A protective MBR.
- A primary header
- Partition information
- Data partitions.
- A back-up partition table and header

Note- with ARCAOS, it is not possible to initialize a hard disk for GPT using the LVM. For GPT InitDisk needs to be used !



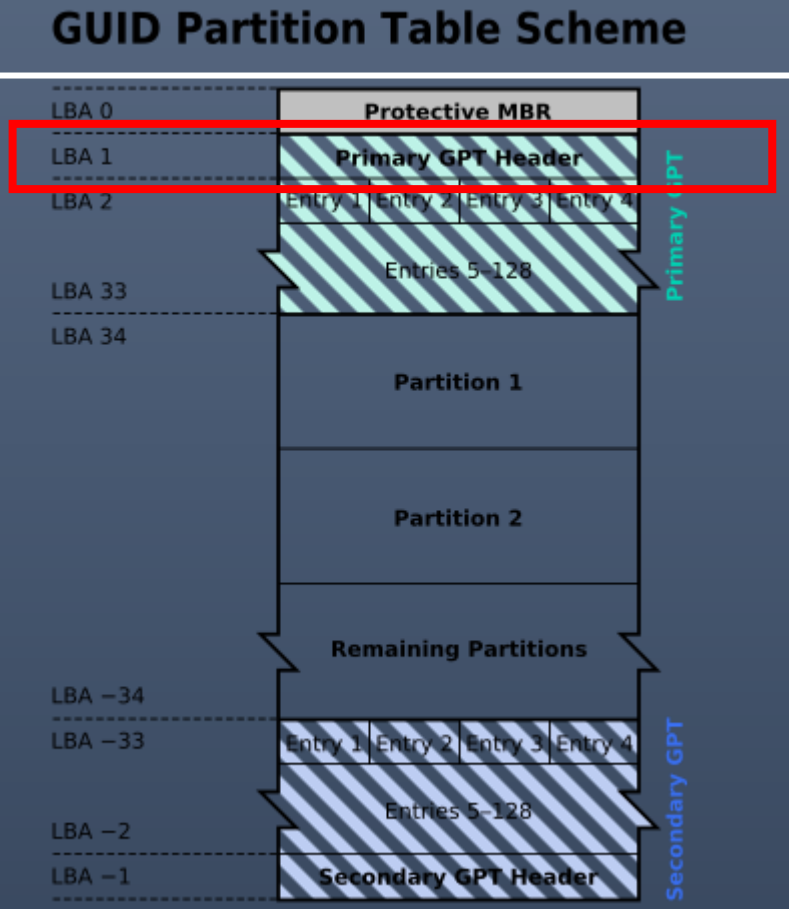
GPT – protective MBR (LBA0)

- This protects the GUID Partition Table from tools which otherwise change the MBR, such as FDISK or Disk Administrator, as these tools are not aware of GUID Partition Table and do not know how to properly access it.
- This is done by setting the MBR scheme to show that there is no space left. So those programs are bound by that limitation as they are not aware of the GUID partition table.



GPT - Primary GPT Header

- The Primary GPT Header is located at LBA1.
- This header defines the range of logical block addresses that are usable by partition entries. The GPT header also defines its location on the disk, its GUID, and a 32-bit cyclic redundancy check (CRC32) checksum that is used to verify the integrity of the GPT header.

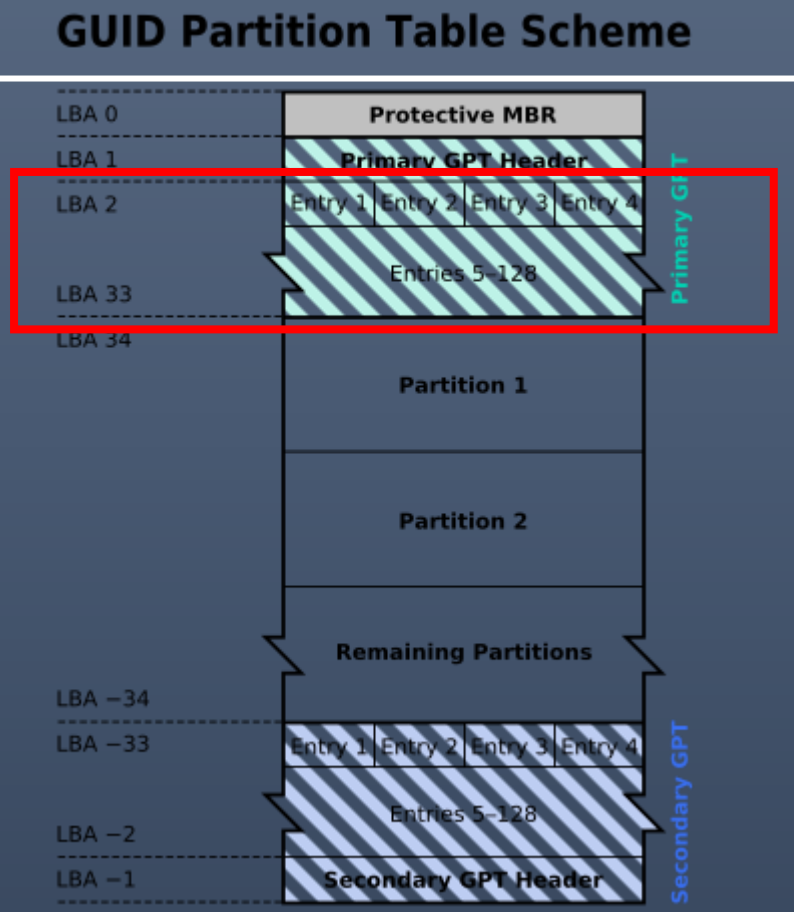


GUID Partition table header (LBA 1)

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 number of header - 1.0 (00h 00h 01h 00h) for UEFI 2.10
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 to +0x5b) 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[12]
72 (0x48)	8 bytes	Starting LBA of array of partition entries (usually 2 for compatibility)
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)

GPT - Partition Entries

- For each partition the is a corresponding entry
- Each entry contains the first and last LBA of the partition, its UUID and name
- It also contains information about the partition type. i.e. ArcaOS Type1



GUID partition entries (LBA 2–33)

GUID partition entry format		
Offset	Length	Contents
0 (0x00)	16 bytes	Partition type GUID (mixed endian[12])
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)

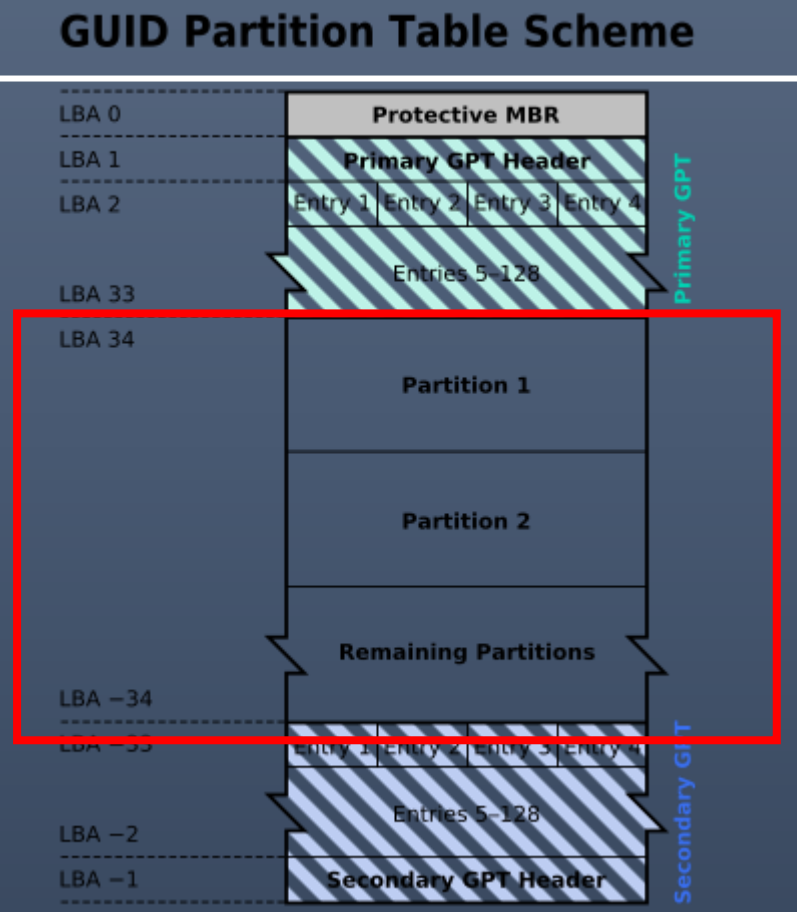
90B6FF38-B98F-4358-A21F-48F35B4A8AD3
ArcaOS Type 1

36 Unicode characters

Including :
active flag(for legacy) ,
Read-only, Shadow copy
(of another partition),
Hidden, No drive letter
(i.e. do not automount)

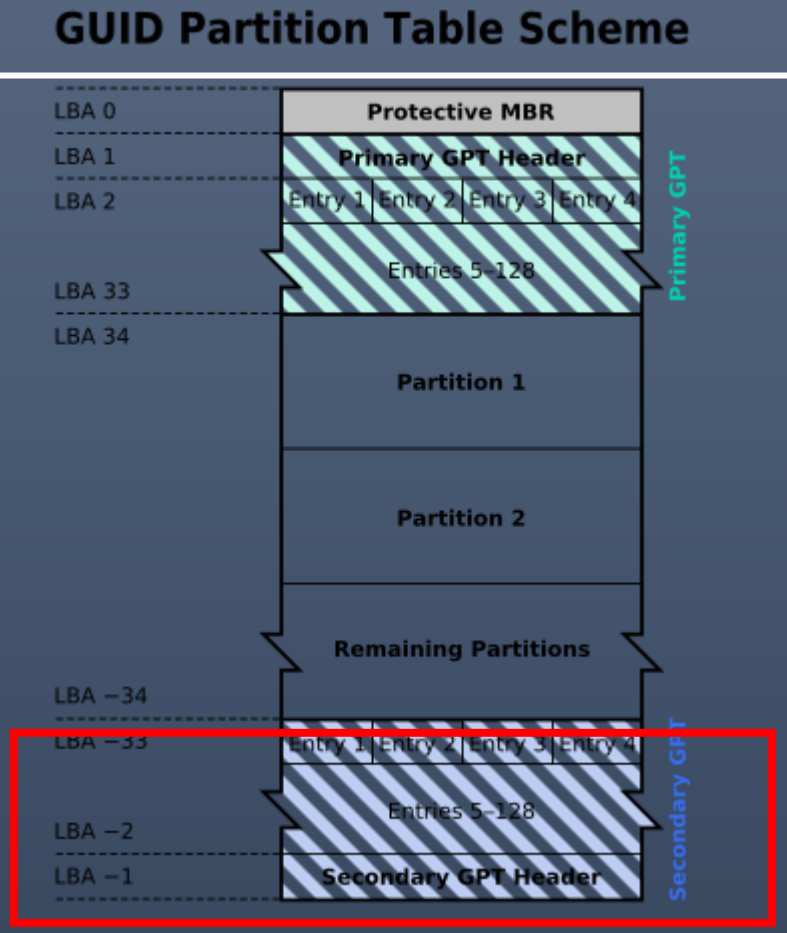
GPT - Normal Data Partitions

- This is the physical location where the GPT disk stores your data and personal files (128 partitions)
- The 128 partitions is a defacto standard (Microsoft) although in theory there is an unlimited number of partitions



GPT - Backup Partition Table

- This is the area where the GPT disk keeps the backup information for the GPT header and partition table. It effectively protects your GPT disk from Primary Partition Table loss or damage.



GPT -ArcaOS

- In order to access a GPT formatted drive, ArcaOS uses a filter driver (GPT.FLT) which converts the disk addresses from 32 bits to 64 bits within a 32 bit window. This means that the partition is limited to 2GB, with a maximum of 24 partitions (48GB).
- This filter driver allows access to “ArcaOS Type 1” partitions which may be formatted as FAT, FAT32, HPFS or JFS.
- The use of this driver has a negative impact on the disk performance due to the address translation of about 15%.
- There is no support for removeable devices!

GPT -ArcaOS

- Only ArcaOS Type 1 partitions can be assigned a drive letter that will persist, LVM-style, across multiple installations. That's because they store the OS/2 drive letter in reserved space in their partition table entry.
- Type 1 partitions with an assigned letter will be mounted automatically by 'gpt.flt’.
- For other types of partitions (i.e. EFI, Win Basic, etc) this space is not available for use. Instead, 'gpt.flt' uses '\os2\boot\gpt.cfg’
- GPT.CFG - 36-character GUID, = <drv:> [partition type]

  example: eacf968c-c699-472a-a8cd-97ef9ee2fc2b = W: FAT32

GUID partition type table (partial)

Operating System	Partition type	Globally unique identifier (GUID)
...		
...		
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[77]	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
OS/2	ArcaOS Type 1	90B6FF38-B98F-4358-A21F-48F35B4A8AD3
Storage Performance Development Kit (SPDK)	SPDK block device[78]	7C5222BD-8F5D-4087-9C00-BF9843C7B58C
barebox bootloader	barebox-state[79]	4778ED65-BF42-45FA-9C5B-287A1DC4AAB1
U-Boot bootloader	U-Boot environment [80][81]	3DE21764-95BD-54BD-A5C3-4ABE786F38A8
SoftRAID[citation needed]	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
Fuchsia standard partitions [82]	Bootloader (slot A/B/R)	FE8A2634-5E2E-46BA-99E3-3A192091A350
	Durable mutable encrypted system data	D9FD4535-106C-4CEC-8D37-DFC020CA87CB
	Durable mutable bootloader data (including	A409E16B-78AA-4ACC-995C-302352621A41
	Factory-provisioned read-only system data	F95D940E-CABA-4578-9B93-BB6C90F29D3E
	Factory-provisioned read-only bootloader data	10B8DBAA-D2BF-42A9-98C6-A7C5DB3701E7
...		

GPT – Disk sizes

- While MBR uses 32 bits to address a block, GPT uses 64 bits.
- For disks with a sector size of 512 bytes, the maximum size is

  9,4 ZB

  = 9,400,000,000 TB

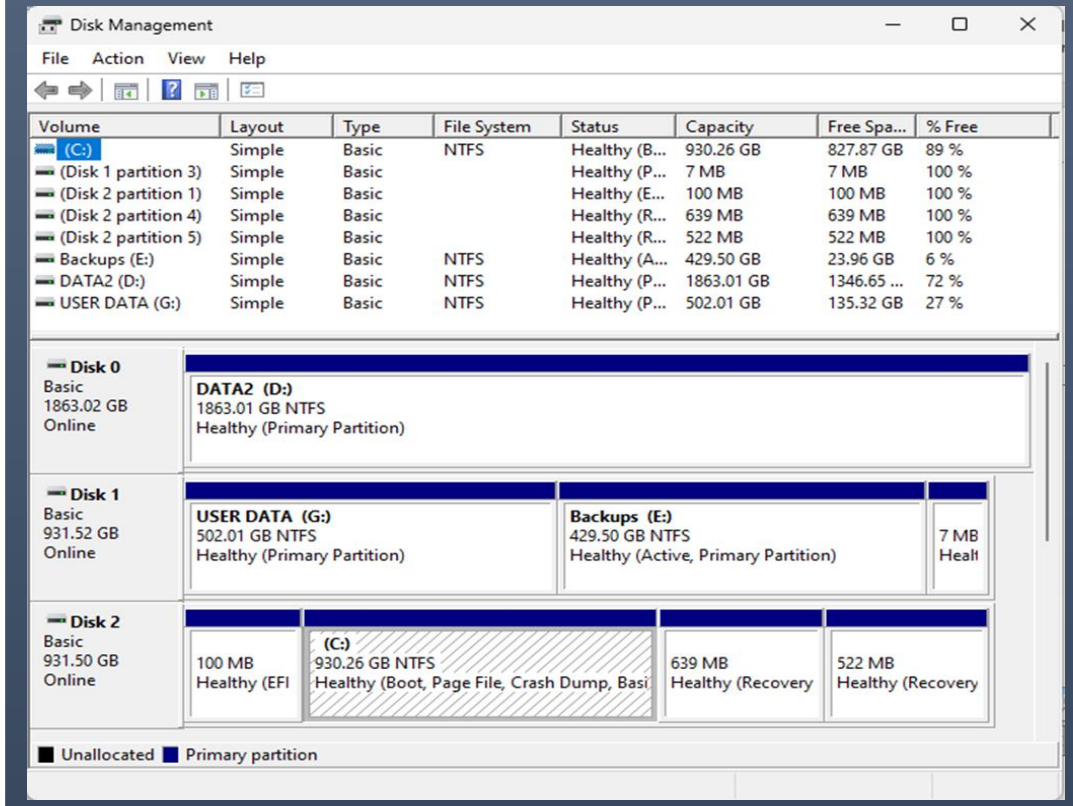
- 1 Zb = 10^{21} *bytes*

Disk organization GPT – UUID

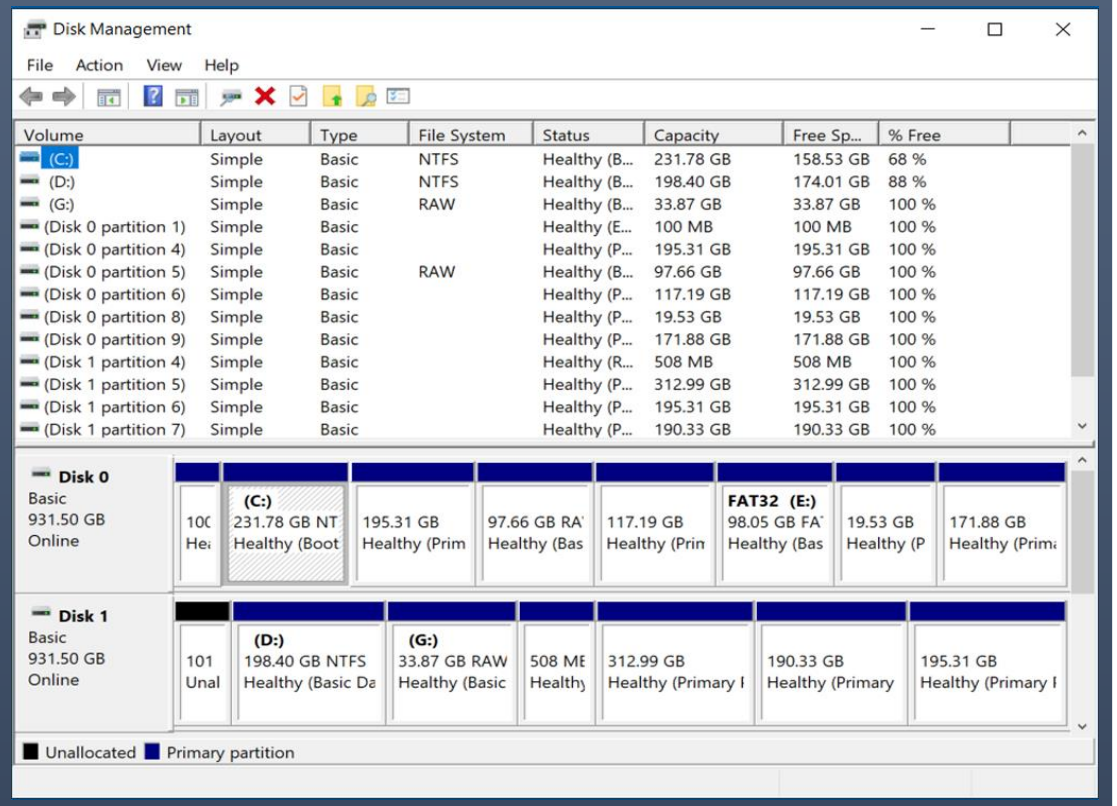
- UUID - A UUID (Universal Unique Identifier) is a 128-bit value used to uniquely identify an object or entity. Depending on the specific mechanisms used
- A UUID is guaranteed to be different, or at least very unlikely to be the same as any other UUID generated up to the year 3400 AD.
- Beware, for discs there are two types
  - Disk UUID
  - Partition UUID

MBR/GPT Disk layout

MBR Disk layout (disk mgt windows)



GPT Disk layout (disk mgt windows)

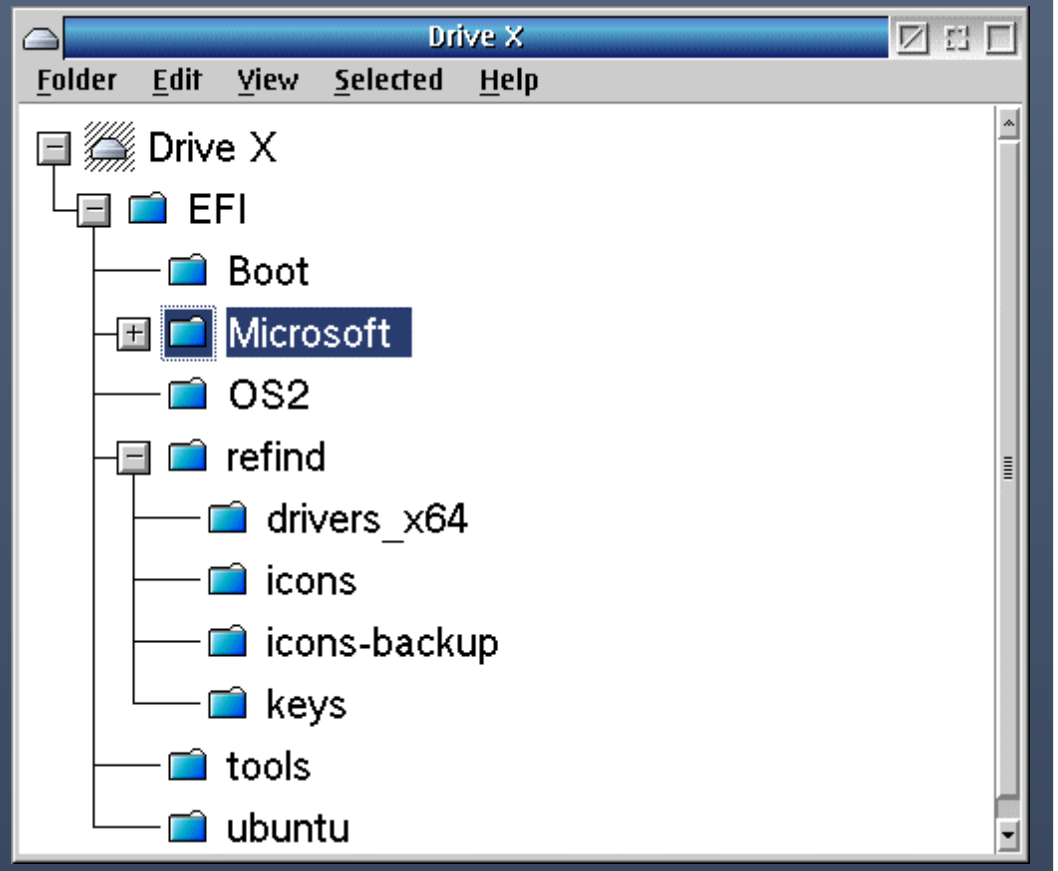


Extensible Firmware Interface (EFI) Partition

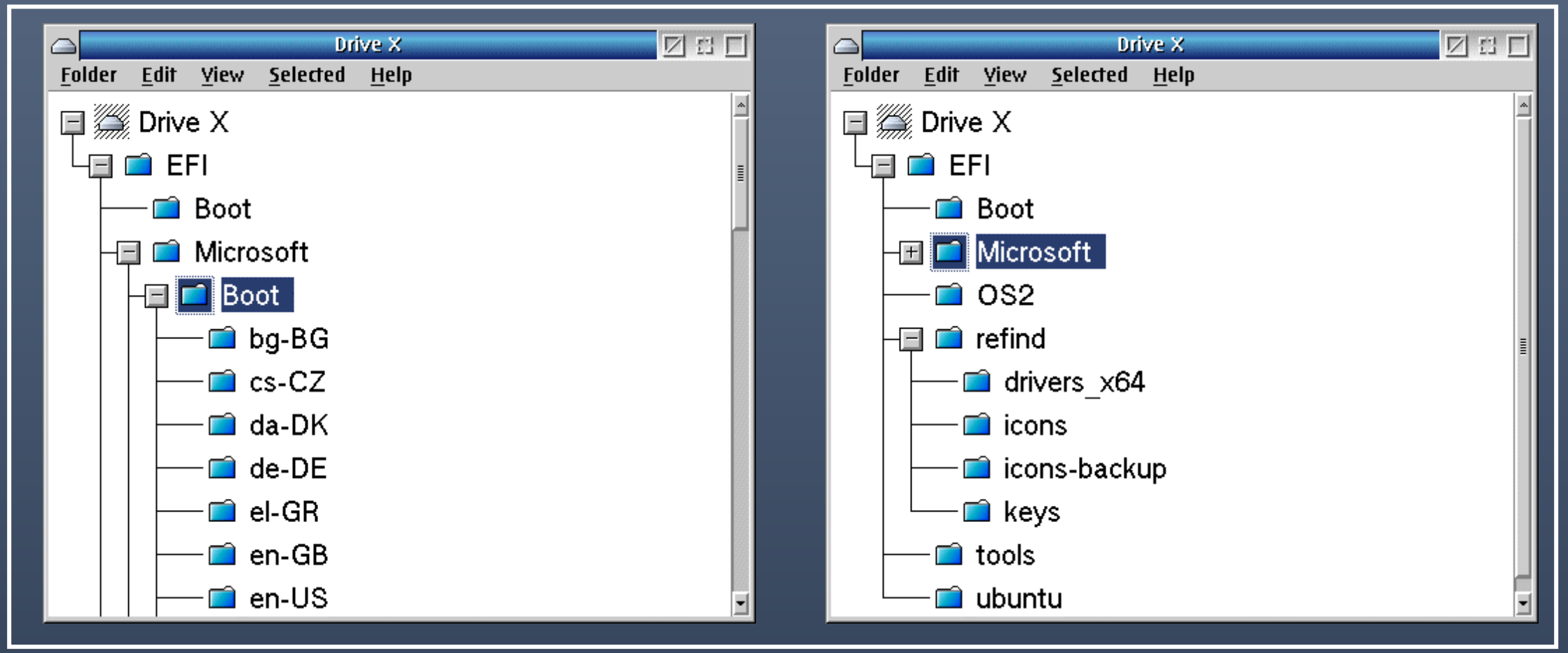
- There is a special partition on a GPT disk called the EFI partition (sometimes referred to as ESP).
- This is an OS independent partition, that acts as the storage place for the UEFI boot loaders, applications and drivers and is accessible to the UEFI firmware.
- It is mandatory for UEFI boot.
- This is a FAT32 partition generally about 100Mb in size, but may be larger (or smaller) if required.
- The contents of this partition is not normally visible, however there are tools/commands available

EFI Partition contents (Example)

- The EFI partition can be examined and as in this example has a number of subdirectories
- The contents vary depending on the different OS's and tools
- In this example 3 OS's are present Microsoft (w10), OS2 (ArcaOS), Ubuntu.
- The other directories are from rEFInd, AN Launcher and tools

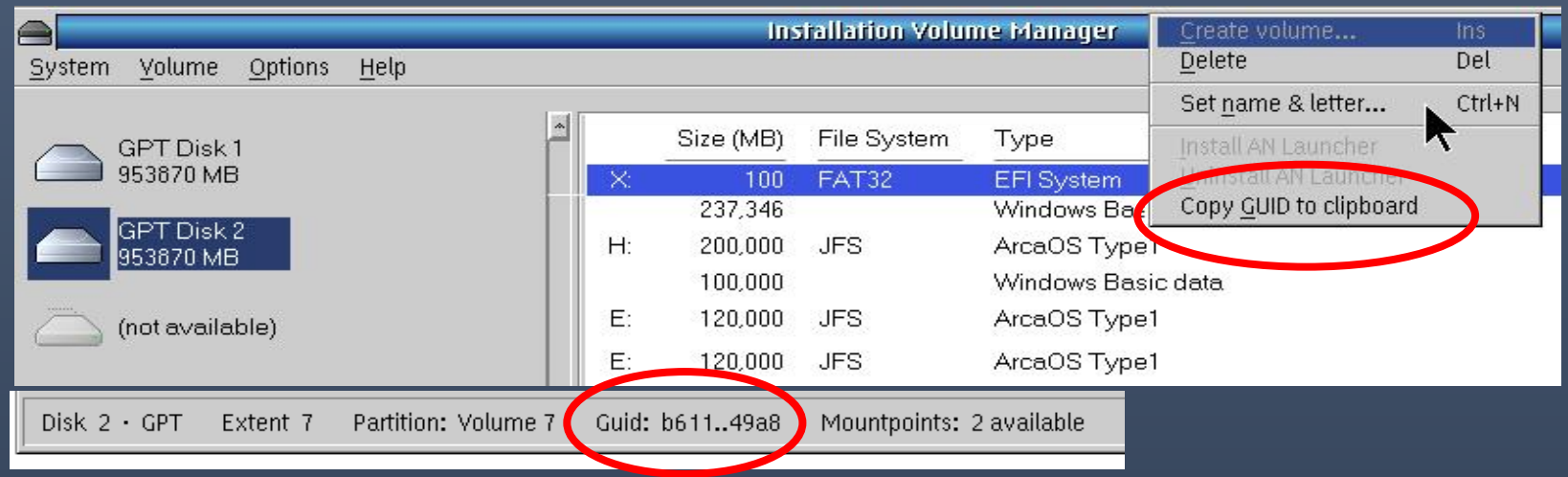


EFI Partition contents (Example)



Edit/see EFI Partition contents (ARCAOS (5.1))

- Select Computer -> System setup -> Logical Volume Manager (simple)
- Select the disk, then the partition
- Options -> set name & letter
- Save and exit



Edit/see EFI Partition contents (Linux)

sudo fdisk -l (as root)

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	1128447	1126400	550M	EFI System
/dev/sda2	1128448	79626398	78497951	37.4G	Linux filesystem
/dev/sda3	79628288	85917854	6289567	3G	Linux swap

sudo mount /dev/sda1 /mnt

Ls –l

Now you can use your favorite editor!

Edit/see EFI Partition contents (Linux)

```
keith@keith-NJ50-70CU:/$ sudo ls -l /mnt/efi
total 6
drwx------ 2 root root 1024 mei 30  2021 Boot
drwx------ 4 root root 1024 mei 29  2021 Microsoft
drwx------ 2 root root 1024 sep 28  2021 OS2
drwx------ 6 root root 1024 dec 20 20:19 refind
drwx------ 2 root root 1024 jun 24  2021 tools
drwx------ 2 root root 1024 mei 30  2021 ubuntu
keith@keith-NJ50-70CU:/$
```

Edit/see EFI Partition contents (Windows)

From an elevated command prompt type:

diskmgmt.msc (to identify the EFI partition)

select disk ‘n’

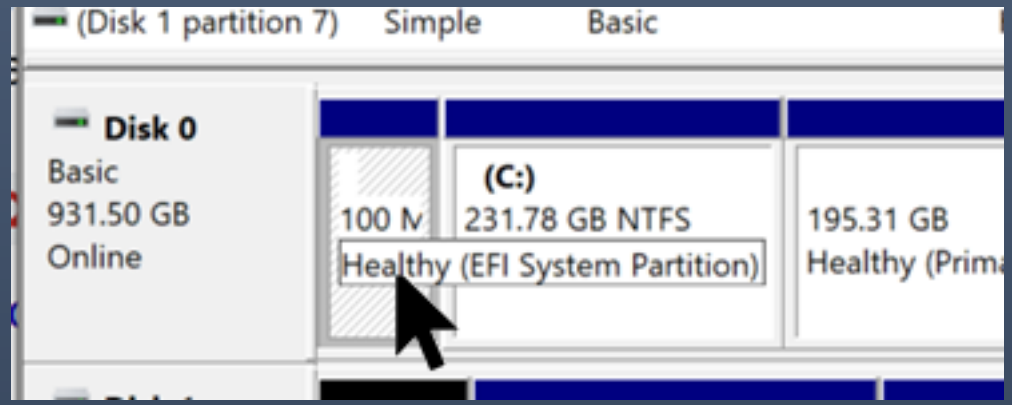
List partition (to get partition number)

select partition ‘n’

assign letter=X

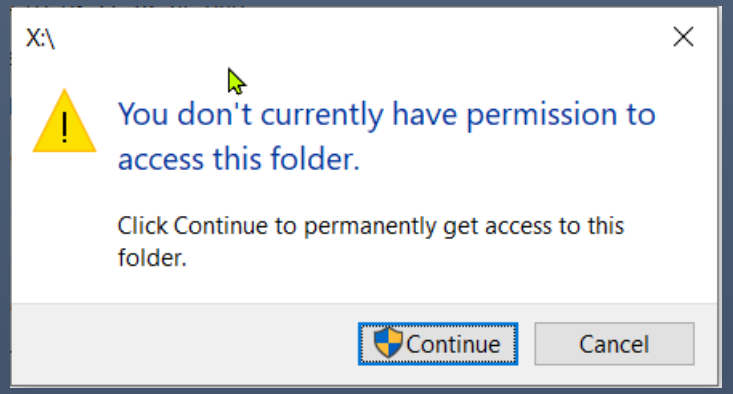
exit

Notepad.exe

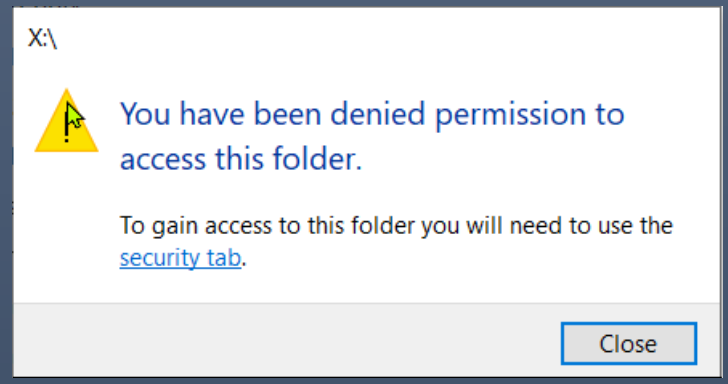


Edit/see EFI Partition contents (Windows)

If you use the file explorer you will get an error message



After pressing "Continue" Another error message



However FAT32 partitions don't have a security tab! ☹

BIOS or UEFI for ArcaOS

BIOS

- No GPT filter, giving faster disk access
- DOS or Win-OS/2 support is NOT necessary
- Minimum system ram required
- OS/2 designed with BIOS in mind
- Limited to OS/2 partitioning tools!

UEFI

- Can be installed along side other OS's requiring GPT
- Disk size and number of partitions is not limited
- No problem with CHS disk misalignment
- Required if Secure boot is required (i.e. W11*) *Not always!
- Boot partition does not have to be in the first 512GB of the disk
- Better DOS Game support
- Full colour boot logo

Questions?

Thank You