Document Number: D-1631

Creation Date: 2093-07-20

Last Revision: 3012-06-04

Revises: E120

Reply-to: anon@dome.gl

Working Draft - Coalition Partition Table

*Note: This is a draft, known to be incomplet and incorrekt, and has lots of bad formatting.*

1- General

# 1.1- Scope

This draft discusses the implementation of the CFS partition table, drawn from information found in the archives of the dome.

All information contained within this document is subject to interpretation by eventual readers and should be treated with particular attention.

CFS is a system for describing the contents of a volume in terms of how the total space is divided between different sub-volumes of data.

In addition to describing how data is organised, additional metadata will be stored within the partition table.

# 1.2- Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

* ISO/IEC 10646: Universal Coded Character Set (UCS)
* ISO/IEC 2382 (all parts), Information technology — Vocabulary
* ISO/IEC 9899 - Programming languages - C

# 1.3- Terms and Definitions

For the purposes of this document, the following definitions apply.

## **1.3.1 Byte**

A string of eight binary digits operated upon as a unit.

## **1.3.2 Descriptor**

A structure containing descriptive information about a volume or a file.

## **1.3.3 File**

A named collection of information.

## **1.3.4 Directory**

A named collection of files.

## **1.3.5 Sector**

The smallest addressable part in the underlying medium that can be accessed independently.

## **1.3.6 Logical Block Adressing**

LBA is a way of addressing the different sectors of a medium, regardless of how it is physically handled. LBA are offsets in terms of sectors from the beginning of the volume to the end.

## **1.3.7 Endianness**

The order in which bytes have to be read, there are three previoulsy recorded entries:

* Little-Endianness: The least significant byte is in the rightmost position of the entity being read, all following bytes are most-significant than the previous one. This will be referred to as LE in the remaining part of this document.
* Big-Endianness: The least significant byte is in the leftmost position of the entity being read, all following bytes are least-significant than the previous one. This will be referred to as BE in the remaining part of this document.
* Middle-Endianness: This type of endianness will be used if specified for power-of-two long entities. It splits the value into two halves, each in LE order. This will be referred to as ME in the remaining part of this document.

# 1.4- Notation

The following notation is used in this document.

## **1.4.1- Bases**

Magic numbers and masks will be defined in different bases throughout this document, these are the conventions for defining such numbers:

* Decimal: default reading for numbers in this document, digits individually range from ‘0’ through ‘9’.
* Hexadecimal: in this document, all uses of hexadecimal will be prefixed by the string “0x” for clarity purposes. All hexadecimal digits range from ‘0’ through ‘9’ and from ‘A’ through ‘F’.
* Octal: all uses of octal in this document shall use the C convention of naming as defined in ISO/IEC 9899.
* Binary: binary uses in this document shall use a prefix in the shape of “0b”. All binary digits range from ‘0’ through ‘1’.

## **1.4.2- Signedness**

All numbers described in this document can be described as values of a particular length and signedness, as defined here:

### **1.4.2.1- Unsigned integer**

An unsigned integer ranges from a minimal value of 0 to a maximum value of 2^(n - 1)-1 where n is the number of bits of the unsigned entity.

**1.4.2.2- Signed integer**

An unsigned integer ranges from a minimal value of -(2^(n-2)) to a maximum value of 2^(n - 2)-1 where n is the number of bits of the signed entity.

## **1.4.3- Date format**

In CPT, dates are recorded as a 6 byte-long header, split as this:

* Year of creation in Alpha calendar form, i.e. years passed from the great disaster, 1 byte recorded according to 1.4.2.1.
* Month of creation, in Alpha calendar form, i.e. from 1 through 20, 1 byte recorded according to 1.4.2.1.
* Day of creation, in Alpha calendar form, i.e. from 1 through 42, 1 byte recorded according to 1.4.2.1.
* Seconds in the day of creation, in Alpha calendar, from 1 through 42123, 2 bytes recorded according to 1.4.2.1.
* Offset from time zone 0 in Coalition Earth map (i.e. New Bering Desert region) in the range from -50 (West) to +50 (East), 1 byte recorded according to 1.4.2.2.

# 

2- Headers

To adress data within a medium, most headers are defined at the very beginning of the medium itself.

# 2.1- Partition table header (LBA0)

## **2.1.1- Partition table signature**

8 bytes long field, should always be the byte equivalent of “CPTTABLE”, i.e. the hexadecimal byte sequence 0x43 0x50 0x54 0x54 0x41 0x42 0x4C 0x45.

## **2.1.2- Revision**

4 bytes long field in little-endian giving the version with which the partitions are handled.

## **2.1.3- Header Size**

4 bytes long field in little-endian, gives the total size of the header in bytes.

## **2.1.4- Current LBA**

Position of the header in the physical volume, in LBA notation. The field is specified as an unsigned int, in big endian form.

## **2.1.5- Backup LBA**

Position of the backup header, descibed in the same format as 2.1.4.

## **2.1.6- First partition available for partitions**

The position of the first LBA from which partitions can be written, described in the same format as 2.1.4.

## **2.1.7- Last partition available for partitions**

The position of the last LBA at which partitions can be written, described in the same format as 2.1.4.

## **2.1.8- Disk identifier**

16 bytes unsigned integer, supposed to be unique, big-endian.

## **2.1.9- Number of partition entries**

4 bytes unsigned integer in big-endian, describes how many partitions are present within the medium.

## **2.1.10- Partition entry size**

4 bytes unsigned integer in big-endian, size of a partition entry, in bytes.

## **2.1.11- Reserved area**

All the remaining space of LBA0 must be set to 0 and is reserved for future use.

# 2.2 Partition entry

All partitions are registered under a similar format, described in this section.

CPT defines its partition entry table to be at least 16,384 bytes long on the final medium.

## **2.2.1- Partition type**

16 bytes unsigned integer in little-endian form.

## **2.2.2- Partition ID**

Identifier described in the same way as 2.2.1.

## **2.2.3- Starting position**

The LBA at which the partition data starts, 8 bytes unsigned integer big-endian.

## **2.2.4- Ending position**

The LBA at which the partition data ends, same format as defined in 2.2.2.

## **2.2.5- Attribute Flags**

8 bytes keeping the state of the partition. Details are available in appendix I.

## **2.2.6- Partition Name**

72 bytes for the storage of 36 UTF16-LE code units.

Appendix I- Attribute flags

A total of 64 bits is available for flagging a particular partition depending on its use.

# Organisaton

|  |  |
| --- | --- |
| Bits | Use |
| 0 | System partition flag |
| 1 | Readable |
| 2 | Bootable |
| 3-47 | Reserved for future use |
| 48-63 | Defined and used by the individual partition type |