

# **Garmin Image (IMG), UserData (ADM), and Text-to-speech (TTS) Container File Format**

Herbert Oppmann

[herby@memotech.franken.de](mailto:herby@memotech.franken.de)

<http://www.memotech.franken.de/FileFormats/>

2021-12-02

## Content

|  |    |
|--|----|
| Garmin IMG, ADM, and TTS Container File Format ..... | 3  |
| Basic data types .....                               | 4  |
| General file structure .....                         | 4  |
| Boot sector .....                                    | 4  |
| Header .....   | 5  |
| Partition Table Entry .....                          | 7  |
| CHS .....  | 7  |
| File Table .....                                     | 7  |
| File Table Entry .....                               | 7  |
| How to start? .....                                  | 8  |
| Text-to-speech (TTS) Subfiles .....                  | 9  |
| References .....                                     | 10 |
| Used sources of information .....                    | 10 |
| Standards and specifications .....                   | 10 |
| Sources of sample files .....                        | 10 |

## Garmin IMG, ADM, and TTS Container File Format

The Garmin Image (IMG), UserData (ADM), and Text-to-speech (TTS) Container File Format is used to combine a set of files into a single file. It is similar to a file system on a disk, but it lacks hierarchy. The contained files (called subfiles) form a flat list – there are no subdirectories. Therefore, all contained files must have unique file names.

This container format is used for image files (IMG) with filename extension \*.img , for user data files (ADM) with filename extension \*.adm, and for Text-to-speech data which usually appears in \*.RGN files in region 17.

IMG files contain all the info needed to render a map on Garmin GPS units.

ADM files can be created by Garmin HomePort [13] and [14]. The program will only offer this feature when it thinks that a marine GPS set is connected. Either the device needs to be connected via USB, or the SD Memory Card of the device needs to be plugged into a reader on the computer. (Actually, HomePort accepts any drive with a certain directory structure and device identification files on it as a marine GPS device.) The user data files are written into subdirectory Garmin\UserData.

With TTS (text-to-speech) the device also speaks the street names with the directions.

The subfiles encountered in a Garmin container depend on the use case. Here is a list of the filename extensions:

IMG: DEM, GMP, IDX, LBL, MAR, MD2, MDR, MET, MPS, NET, NOD, QSI, RGN, S16, SNR, SRT, TRE, TDB, TRF, and TYP

ADM: PRX , RTE, TRK, and WPT

TTS: BAK, BDC, BIN, DAT, DCT, HDR, SIG, TST, TXT, and VXM

The IMG and ADM subfiles are very complex and also exist as stand-alone files. They are therefore described in separate documents. This document covers the IMG, ADM, and TTS container file format, and the TTS subfiles.

This documentation is based on own research and the sources listed in the references section.

Text highlighted like **this** is where further investigation is needed.

## Basic data types

All values are serialized in little-endian byte order (least significant byte first).

| Type   | Length | Description  |
|--------|--------|--|
| char   | 1      | A byte, which is a character code according to [15]. |
| byte   | 1      | 8 bit unsigned integer (range 0 .. 255)              |
| ushort | 2      | 16 bit unsigned integer (range 0 .. 65535)           |
| uint   | 4      | 32 bit unsigned integer (range 0 .. 4294967295)      |

Compressed part number (see [9]):

The 9x8 bits have to be interpreted as 12 character codes with 6 bits each.

Example: 10 D4 40 56 14 91 0D 14 41

Reverse bytes (shown in binary format with most significant bit (bit 7) first):

0100 0001 0001 0100 0000 1101 ...

Create groups of six bits each:

010000 010001 010000 001101 ...

Add 0x20 = 0010 0000, this may overflow into the 7th bit:

00110000 00110001 00110000 00101101 ...

Interpret bytes as ASCII code:

'0' '1' '0' '-' ...

## General file structure

|   |
|---|
| Boot sector   |
| Optional: One or more gap sectors, filled with 0x00 |
| File table  |
| Clusters of the contained files                     |

The boot sector, the gap sectors, and the file table, together are treated also as a special type of file. It appears in the file table with a name consisting of all blanks (' '). I named this CTRL for control. All contained files, including CTRL, are managed in cluster granularity, which is a power of 2 times the sector size (512 byte). Within the CTRL file, all content (boot sector, gap sectors, file table and their entries) starts on sector boundaries.

## Boot sector

This sector is similar to a Master Boot Record (MBR, see [10]).

| Offset      | Content                                |
|-------------|--|
| 0x000-0x088 | Header                                 |
| 0x089-0x1BD | unused, = 0                            |
| 0x1BE-0x1CD | Partition Table Entry 1                |
| 0x1CE-0x1FE | Partition Table Entry 2-4, unused, = 0 |
| 0x1FE-0x1FF | MBR signature, = 0x55 0xAA             |

## Header

| Offset | Type    | Content  |
|--------|---------|--|
| 0x00   | byte    | XOR byte<br>If this byte is $\neq 0$ , the container file is obfuscated. Each byte of the complete file must be XORed with this value for getting the plain data.<br>In theory, the file always has to be XORed with this value, but a value of 0 will not change anything.  |
| 0x01   | byte[7] | = 0  |
| 0x08   | byte    | Map version major  |
| 0x09   | byte    | Map version minor  |
| 0x0A   | byte    | Map update month, 1..12 (this is what HomePort uses, but some tools obviously use 0..11)   |
| 0x0B   | byte    | Map update year<br>MapEdit++ adds 1900 for val > 90, else 2000.<br>The vast majority of dates > 2000 use values >100 here, only very few from the early 200x years use low numbers. To me, it seems that the low numbers came from some buggy implementation, and the rule is only there for compatibility to this bug.<br>Most implementations just add 1900 regardless of the value. |
| 0x0C   | ushort  | =0   |
| 0x0E   | byte    | Map source flag<br>0=Garmin map visible in Basecamp and Homeport<br>1=file created by Mapsource<br>TTS has always 0 here, ADM has always 1 here  |
| 0x0F   | byte    | Byte checksum<br>0=not set<br>otherwise: The (byte) sum of all bytes of the file, including the checksum byte, shall be 0.<br>ADM has always 0 here.   |
| 0x10   | byte[6] | Signature1 "DSKIMG", or "DSDIMG" for a demo map.<br>(Some sources mention "DSKDEM", but if this is true, it is very old. Available firmwares contain the strings "DSKIMG", "DSDIMG" and "GARMIN" directly in sequence, but do not contain "DSKDEM".)   |
| 0x16   | ushort  | Sector size, =512  |
| 0x18   | ushort  | Sector1. Values seen: 4, 8, 16, and 32   |
| 0x1A   | ushort  | Head1. Values seen: 16, 32, 64, 128, and 256   |
| 0x1C   | ushort  | Cylinder1. Values seen: 32, 256, 512, 1023, 1024, and also odd values  |
| 0x1E   | ushort  | =0   |
| 0x20   | uint    | =0   |
| 0x24   | uint    | uint checksum<br>0=not set<br>otherwise: The (uint) sum of all bytes of the file (skipping the byte checksum at offset 0x0F and skipping this uint checksum).<br>Is != 0 only on maps produced by Garmin.<br>ADM has always 0 here.  |
| 0x28   | uint    | =0   |
| 0x2C   | uint    | =0   |

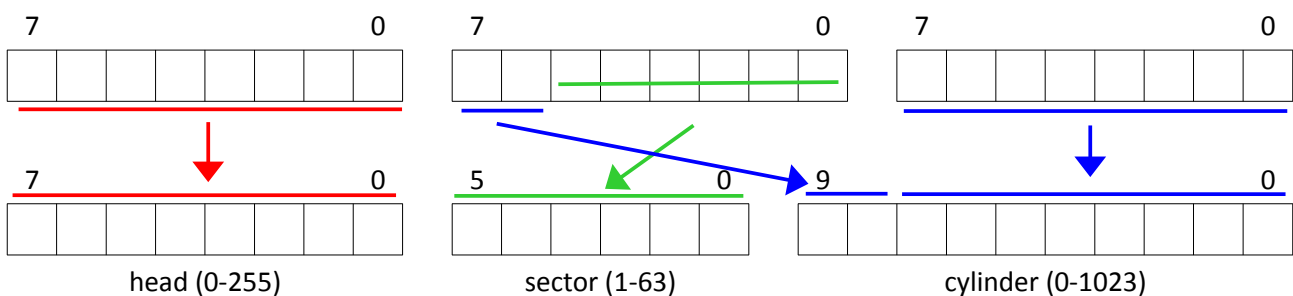
| Offset | Type     | Content   |
|--------|----------|---|
| 0x30   | byte[9]  | Compressed part number  |
| 0x39   | ushort   | Creation year   |
| 0x3B   | byte     | Creation month, 1-12 (this is what HomePort uses, but some tools obviously use 0..11)   |
| 0x3C   | byte     | Creation day, 1-31  |
| 0x3D   | byte     | Creation hour, 0-23   |
| 0x3E   | byte     | Creation minute, 0-59   |
| 0x3F   | byte     | Creation second, 0-59   |
| 0x40   | byte     | Sector number where file table starts within CTRL file. Values seen: 2 and 8 (maybe depends on cluster size)  |
| 0x41   | byte[7]  | Signature 2 "GARMIN\0"  |
| 0x48   | byte     | =0, Maybe the Signature2 is one byte shorter and this is a ushort   |
| 0x49   | char[20] | Map description, ADM has "USERDATA" here, TTS has the name of the "speaker" here in all lower case letters or just the first letter capitalized.<br>In IMG and TTS it is always padded with space (0x20) chars - in ADM it is padded with space or 0x00.  |
| 0x5D   | ushort   | Head2, == Head1   |
| 0x5F   | ushort   | Sector2, == Sector1   |
| 0x61   | byte     | Bytes per Sector exponent, =9 for 512 bytes per sector  |
| 0x62   | byte     | Sectors per Cluster exponent. Known values: 0..7<br>The cluster exponent must be chosen high enough so that the total number of clusters is < 65535.  |
| 0x63   | ushort   | Total number of clusters in the container file. This should be file size / cluster size, but often, it is either one more or it is even the next power of 2.  |
| 0x65   | char[30] | Map description (continued), ADM and TTS have only padding bytes here.<br>In IMG and TTS it is always padded with space (0x20) chars - in ADM it is padded with space or 0x00.  |
| 0x83   | ushort   | Magic value to tell if the next three fields are valid.<br>0: not valid (field values shall then be 0)<br>0xDEAD: valid   |
| 0x85   | byte     | Version major   |
| 0x86   | byte     | Version minor   |
| 0x87   | ushort   | Release number? For the same product, the value increases steadily ..<br><br>Seen values: 0 when magic value is 0,<br>else: 0, 9, 0x000B, 0x0024, 0x0047, 0x0056, 0x0058, 0x0062, 0x008B, 0x00A5, 0x00E5, 0x00E6, 0x00EB, 0x00F9, 0x00FC, 0x0117, 0x0134, 0x0136, 0x0138, 0x013A, 0x013B, 0x013F, 0x0144, 0x0146, 0x0152, 0x0153, 0x0159, 0x015A, 0x015C, 0x0183, 0x01A9, 0x0228 and 0x05B2 |

BLUCHART maps have something different in their checksum calculation ...

## Partition Table Entry

| Offset | Type | Content   |
|--------|------|---|
| 0x1BE  | byte | Status: 0x00=non-bootable, 0x80=bootable, =0  |
| 0x1BF  | CHS  | CHS address of first absolute sector in partition, = 0/1/0  |
| 0x1C2  | byte | Partition type, =0x00   |
| 0x1C3  | CHS  | CHS address of last absolute sector in partition<br>Cylinder * Head1 * Sector1 + Head * Sector1 + Sector == Number of sectors in partition<br>Actually, this number is nonsense, because on a real disk you don't know Head1 and Sector1, and it is the Cylinder x Head x Sector which defines the size of the disk. I.e. Cylinder is the last usable cylinder, Head is the last usable head, and Sector is the last usable sector, and the partition always covers the full array. |
| 0x1C6  | uint | LBA of first absolute sector in partition = 0   |
| 0x1CA  | uint | Number of sectors in partition. == Cluster * 2 ^ sectors per Cluster exponent.  |

## CHS



## File Table

Note: The file table starts on a sector boundary, not necessarily on a cluster boundary. Most of the times it ends at a cluster boundary, having unused entries at the end if needed. But sometimes it has less entries and there is unused space in the last cluster.

## File Table Entry

Each entry in the file table is one sector in size.

| Offset | Type    | Content  |
|--------|---------|--|
| 0x000  | byte    | Flag: 0=unused, 1=used   |
| 0x001  | char[8] | Filename, padded with space (0x20) chars   |
| 0x009  | char[3] | Filename extension, padded with space (0x20) chars   |
| 0x00C  | uint    | File size in byte  |
| 0x010  | byte    | Only valid in the first extent, otherwise set to 0.<br>Special<br>0 = for payload files<br>1 = for the CTRL file in ADM files<br>2 = for payload files in some IMG files |

| Offset | Type        | Content   |
|--------|-------------|---|
|        |             | 3 = for the CTRL file in IMG and TTS files<br>Only valid in the first extent, otherwise set to 0.   |
| 0x011  | byte        | Extent number, starting with 0  |
| 0x012  | byte[14]    | Unused, usually =0x00, but sometimes =0xFF  |
| 0x020  | ushort[240] | List of Cluster numbers. The first cluster in the file is 0. Unused entries (after the end of the file) are 0xFFFF. The list must be contiguous (there may be no holes in it). The entries are really unsigned short (value 50888 has been seen). |

Big files may occupy more than 240 clusters. In this case, there are multiple file table entries for the same file. Usually, all entries for one file are in consecutive order in the file table, but this can't be relied on. The first entry has extent number 0, the second entry has extent number 1, and so on.

All file table entries for the extents have correct flag and file name, but the file size is set only in the first extent. File size is 0 in the following extents.

The file table entries for the CTRL file are special: The file name and extension is all blanks (0x20). The field "Special" of the first extent is 1 for ADM and 3 for IMG and TTS. The following extents have field "Special" as 0. The size must be a multiple of the sector size. Within some IMG files, the first extents of the normal payload files have the field "Special" set to 2.

For the CTRL file, the file size must be a multiple of the sector size (512 byte) and should be a multiple of the cluster size, but sometimes, it is smaller. In this case, the remaining place up to the end of the cluster is filled with 0xFF.

## How to start?

OK. After we have read in the CTRL file, we know which files are present, their names, how long they are, and where their clusters are. But wait a minute - the CTRL file is itself a file which is described in the CTRL file. So this leads to a chicken and egg problem!

This is solved by the following rules:

- The CTRL file always starts with the first cluster (cluster number 0). So the very first bytes of the IMG/ADM/TTS file are the Header, where we can find out the cluster size and the starting sector of the file table. This linear mapping means that CTRL file offsets are identical to IMG/ADM/TTS file offsets.
- If the file table does not start within the first cluster, all clusters of the CTRL file up to and including the first file table entry must be in consecutive order within the IMG/ADM/TTS file, i.e. the linear mapping continues. This ensures we can read the first file table entry without knowing the file table in advance.
- The file table entry for the first extent of the CTRL file is the very first entry within the file table. This ensures that after we have read the cluster that contains the beginning of the file table, we know at least the first 240 cluster of the CTRL file. In case the CTRL file is so big that it uses more than one extent, the file table entry for the next extent must be within the first 240 clusters of the CTRL file, and so on. This ensures we can step by step determine the CTRL file.

In reality, things are much easier:

Usually the CTRL file starts with the beginning of the IMG/ADM/TTS file and runs contiguously until its end. After that, the first payload file starts and runs contiguously until its end, and so on. So in practice, you will only see consecutive cluster numbers within the file table entries.

So if you want to cut down on effort, you may take some shortcuts in the dissection logic here. You could access the payload file data with just knowing the length and a single offset for each file.

## Text-to-speech (TTS) Subfiles

See e.g. [16].

|             |  |
|-------------|--|
| DLLINFO.TXT | XML, surrounded by "START" and "END"<br>Should be the first file (at least in one case it is the last) |
| *.HDR       | XML  |
| *.DAT       | binary data  |
| DICT.BDC    |  |
| DLLIMG.BIN  | code   |
| DLLIMG.SIG  | binary data, optional  |
| *.VXM       | XML  |
| *.TXT       | text in UTF-8, sometimes with BOM  |
| *.DCT       | text in UTF-8  |
| *.DCT.BAK   | backup of *.DCT files  |
| IMGEND.TST  | text "END"<br>Should be the last file (at least in one case, "DLLINFO.TXT" comes after it)             |

## References

### Used sources of information

- [1] libgarmin <http://libgarmin.sourceforge.net/>
- [2] mkgmap <http://www.mkgmap.org.uk/>
- [3] GPSTabel <https://www.gpsbabel.org/>
- [4] Garmin IMG Format <https://sourceforge.net/projects/garmin-img/>
- [5] <http://pinns.co.uk/osm/>
- [6] [http://wiki.openstreetmap.org/wiki/OSM\\_Map\\_On\\_Garmin/IMG\\_File\\_Format](http://wiki.openstreetmap.org/wiki/OSM_Map_On_Garmin/IMG_File_Format)
- [7] [http://whiter.brinkster.net/raster\\_img.shtml](http://whiter.brinkster.net/raster_img.shtml)
- [8] <https://www.gpspower.net/garmin-discussions/224115-lock-img-file.html>
- [9] <https://www.gpspower.net/garmin-maps/266563-garmin-transalpin-2012-pro-10.html>
- [10] [https://en.wikipedia.org/wiki/Master\\_boot\\_record](https://en.wikipedia.org/wiki/Master_boot_record)
- [11] Decoding Proprietary Marine Track Files <https://github.com/rahra/parsefsh>
- [12] Perl script which converts user data to industry-standard GPX records  
<https://sourceforge.net/projects/adm2gpx/>
- [13] Garmin HomePort [https://www8.garmin.com/support/download\\_details.jsp?id=7263](https://www8.garmin.com/support/download_details.jsp?id=7263)
- [14] "Send an ADM file to an SD card for a chartplotter using HomePort" <https://support.garmin.com/en-US/?faq=f5VyBm3i5s9P46OJsV6RbA>

### Standards and specifications

- [15] ISO/IEC 8859-1:1998 Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1

### Sources of sample files

- [16] [http://download.garmin.com/software/Deutsch\\_Yannick\\_D4481-00\\_0210.rgn](http://download.garmin.com/software/Deutsch_Yannick_D4481-00_0210.rgn), Region 17
- [17] [http://wiki.openstreetmap.org/wiki/OSM\\_Map\\_On\\_Garmin/Download](http://wiki.openstreetmap.org/wiki/OSM_Map_On_Garmin/Download)
- [18] <http://www.garniak.pl/viewtopic.php?t=398>
- [19] [https://www.digital-eliteboard.com/forums/seekarten.976/?prefix\\_id=22](https://www.digital-eliteboard.com/forums/seekarten.976/?prefix_id=22)
- [20] Desert Southwest Map <http://www.gpsfiledepot.com/maps/view/518/>
- [21] Bucktail path  
[http://www.zinnware.com/HighAdv/BucktailPath\\_200305/GPS\\_tracking\\_download.html](http://www.zinnware.com/HighAdv/BucktailPath_200305/GPS_tracking_download.html)
- [22] <http://www.switchbacks.com/nwtrails/>
- [23] <http://www.island-olaf.de/travel/marokko/gps.html>
- [24] <http://www.elgps.com/mapas.html>

- [25] <http://www.pltopo.pl/podglad-mapy/>
- [26] [http://www.garda-gps.de/dl\\_pejo.html](http://www.garda-gps.de/dl_pejo.html)
- [27] [https://static.garmin.com/shared/de/custom/topod2012/testversion/GarminTransAlpin2012Pro\\_Testversion.zip](https://static.garmin.com/shared/de/custom/topod2012/testversion/GarminTransAlpin2012Pro_Testversion.zip)
- [28] [https://static.garmin.com/shared/de/custom/topod2012/testversion/TopoD2012pro\\_Testversion.zip](https://static.garmin.com/shared/de/custom/topod2012/testversion/TopoD2012pro_Testversion.zip)
- [29] <https://web.archive.org/web/20190613150155/http://www.bgroutingmap.com/download/BGDemoMapSofia.exe>
- [30] Moscow routing map <http://garminmap.narod.ru/>
- [31]