

Garmin UserData (ADM) Subfiles File Format

Herbert Oppmann

herby@memotech.franken.de

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

2019-11-19

Content

Garmin UserData (ADM) Subfiles File Format	3
Basic data types	3
Common basic structure	4
Main record content (type=0)	4
Checksum record content (type=1)	4
Data record content (type=2) common basic structure	5
PRX subfile data record	6
Proximity descriptor table	6
Proximity records	7
RTE subfile data record	8
Route descriptor table	8
Route point descriptor table	8
Routes	9
TRK subfile data record	10
Track descriptor table	10
Trackpoint descriptor table	10
Tracks	11
WPT subfile data record	12
Waypoint descriptor table	12
Waypoints	12
Enumerations	13
References	21
Used sources of information	21
Standards and specifications	21
Sources of sample files	21

Garmin UserData (ADM) Subfiles File Format

This document contains info about all files contained within ADM files. For the IMG/ADM Container File Format, there is a separate document available.

All subfiles have the name "USERDATA" and only differ in the file extension.

File types (determined by file name extension):

PRX Proximity records

RTE Routes

TRK Tracks

WPT Waypoints

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
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)
ulong	8	64 bit unsigned integer (range 0 .. 18446744073709551615)
short	2	16 bit signed integer (range -32768 .. 32767)
int	4	32 bit signed integer (range -2147483648 .. 2147483647)
float	4	32 bit floating point value according to [5].
long	8	64 bit signed integer (range -9223372036854775808 .. 9223372036854775807)
char	1	A byte, which is a character code according to [6].

fixstring:

String with known fixed length. If the actual string is shorter, it is filled up to the fixed length with zeroes.

Note: The string is not terminated, e.g. by a 0-character.

nfloat:

A float, where the value 1e25 means "no value given".

ndate:

An uint, where the values 0 or 0xFFFFFFFF both mean "no value given". All other values have to be interpreted as unix time (see [3]), but with a starting value (epoch) of 1989-12-31 instead of 1970-01-01.

Note: While the number counts seconds, unix time is not defined as "number of seconds since the epoch", but "number of days since the epoch" x 24x60x60 + "number of seconds since midnight". The difference between the two is that unix time ignores leap seconds. The difference between unix time and UTC is already more than 20 seconds.

MapUnit:

An int, whose value must be multiplied with $360.0^\circ / (1 \ll 32)$ to get a coordinate (latitude or longitude).

Common basic structure

All files contained within an ADM file share the same basic structure:

The file consists of two records directly following each other, the main record and the checksum record. The main record has another record embedded – the data record. Each of the records follows the TLV pattern (type-length-value):

Offset	Type	Content
0	ushort	Record type 0 = main record 1 = checksum record 2 = data record
2	uint	Length of the record in byte. The length includes the type and length fields, so the minimum value is 6 for an empty record.
6	[depends on record type]	Value (content) of the record.

The main record and the checksum record always have the same structure across all subfile types. The data record is different, but very similar in structure.

The records contain some pointer values. Because they only point to data within the same record, one would expect that they are offsets relative to the beginning of the record, but this is not the case. These pointers are absolute file offsets, i.e. relative to the beginning of the file.

Main record content (type=0)

Offset	Type	Content
6	byte	Subfile type 0 = WPT 1 = PRX 2 = RTE 3 = TRK
7	uint	Pointer to embedded data record, typically =15
11	uint	? Values seen: 100 (maybe this means file format version 1.0)
15		The embedded data record usually follows immediately here and uses up all remaining bytes of the main record. Its content depends on the subfile type.

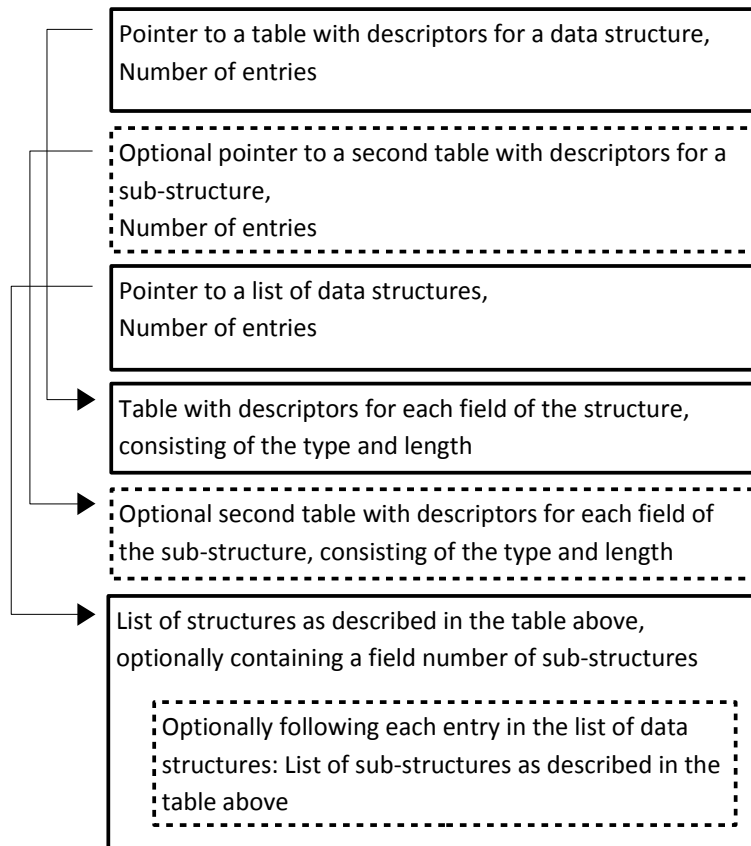
Checksum record content (type=1)

Offset	Type	Content
6	uint	Checksum

The checksum algorithm initializes the sum with a value of zero and individually adds all bytes of the main record, i.e. up to the beginning of the checksum record.

Data record content (type=2) common basic structure

While the data record content is different for the different ADM subfile types, there is a common pattern:



While the tables and lists are all referenced by pointers, in all sample files seen they all follow each other in this order without gaps.

Note: I use the term “table” when all entries have same size, and “list” when the entries have variable size.

PRX subfile data record

Contains all proximity records.

Offset	Type	Content
6		Proximity descriptor table info
	uint	Pointer
	uint	Number of table entries
14		Proximity records info
	uint	Pointer
	uint	Number of proximity records

Proximity descriptor table

The proximity descriptor table usually follows the fixed fields described above. It defines all the fields that make up one proximity record.

Each entry has the following structure:

Type	Content
ushort	Field type (see below)
ushort	Field length

Field types:

Field type	Length	Datatype of value	Meaning
0	1		?
1	4	MapUnit	Latitude
2	4	MapUnit	Longitude
20	0 if number of tracks =0, else >0	fixstring	Name
21	<i>variable</i>	fixstring	Notes
22	2		Symbol, see table below
23	1	byte	Display mode, see table below
24	4	nfloat	Depth in meter
25	4	nfloat	Temperature in °C
26	4	ndate	TimeStamp
40	0 if number of tracks =0, else >0	fixstring	? identical to Name
41	<i>variable</i>	fixstring	Street address
42	<i>variable</i>	fixstring	City
43	2		? is 0
44	1		? is 0
60	4		? is 0
61	4		? is 0xFFFFFFFF
62	8		? is 0xFFFFFFFFFFFFFFFF
63	1		? is 1

Field type	Length	Datatype of value	Meaning
100	4	nfloat	Proximity

Proximity records

The table of proximity records usually follows the proximity descriptor table. Each proximity record has a fixed-length structure as described in the Proximity Descriptor Table.

RTE subfile data record

Contains all routes.

Offset	Type	Content
6		Route descriptor table info
	uint	Pointer
	uint	Number of table entries
14		Routepoint descriptor table info
	uint	Pointer
	uint	Number of table entries
22		Routes info
	uint	Pointer
	uint	Number of routes

Route descriptor table

The route descriptor table usually follows the fixed fields described above. It defines all the fields that make up one route (without its routepoints).

Each entry has the following structure:

Type	Content
ushort	Field type (see below)
ushort	Field length

Field types:

Field type	Length	Datatype of value	Meaning
200	0 if number of tracks =0, else >0	fixstring	Route name
201	2	ushort	Number of routepoints belonging to this route
202	4	uint	Pointer to the routepoints belonging to this track

Routepoint descriptor table

The routepoint descriptor table usually follows the route descriptor table. It defines all the fields that make up one routepoint.

Each entry has the following structure:

Type	Content
ushort	Field type (see below)
ushort	Field length

Field types:

Field type	Length	Datatype of value	Meaning
0	1		?
1	4	MapUnit	Latitude
2	4	MapUnit	Longitude
20	0 if number of tracks =0, else >0	fixstring	Name
21	<i>variable</i>	fixstring	Notes
22	2		Symbol, see table below
23	1	byte	Display mode, see table below
24	4	nfloat	Depth in meter
25	4	nfloat	Temperature in °C
26	4	ndate	TimeStamp
40	0 if number of tracks =0, else >0	fixstring	? identical to Name
41	<i>variable</i>	fixstring	Street address
42	<i>variable</i>	fixstring	City
43	2		? is 0
44	1		? is 0
45	16		
60	4		? is 0
61	4		? is 0xFFFFFFFF
62	8		? is 0xFFFFFFFFFFFFFFFF
63	1		? is 1
80	10		?
81	4		?
82	4		?
83	1		?

Routes

The list of routes usually follows the routepoint descriptor table. Each route has a fixed-length structure as described in the Route Descriptor Table. Each route is immediately followed by a list of routepoints. The length of the list is given by a field in the route structure. Each routepoint has a fixed-length structure as described in the Routepoint Descriptor Table.

TRK subfile data record

Contains all tracks.

Offset	Type	Content
6		Track descriptor table info
	uint	Pointer
	uint	Number of table entries
14		Trackpoint descriptor table info
	uint	Pointer
	uint	Number of table entries
22		Tracks info
	uint	Pointer
	uint	Number of tracks

Track descriptor table

The track descriptor table usually follows the fixed fields described above. It defines all the fields that make up one track (without its trackpoints).

Each entry has the following structure:

Type	Content
ushort	Field type (see below)
ushort	Field length

Field types:

Field type	Length	Datatype of value	Meaning
300	0 if number of tracks =0, else >0	fixstring	Track name
301	2	ushort	Number of trackpoints belonging to this track
302	1	byte	? Seen values 0, 1, 2 and 3
303	1	byte	Color, see table below
304	4	uint	Pointer to the trackpoints belonging to this track

Trackpoint descriptor table

The trackpoint descriptor table usually follows the track descriptor table. It defines all the fields that make up one trackpoint.

Each entry has the following structure:

Type	Content
ushort	Field type (see below)
ushort	Field length

Field types:

Field type	Length	Datatype of value	Meaning
500	4	MapUnit	Latitude
501	4	MapUnit	Longitude
502	4	ndate	Timestamp
503	4	nfloat	Depth in meter
504	1	byte	? Is always 1 on the first trackpoint, else 0 or 1
505	4	nfloat	Temperature in °C

Tracks

The list of routes usually follows the trackpoint descriptor table. Each track has a fixed-length structure as described in the Track Descriptor Table. Each track is immediately followed by a list of trackpoints. The length of the list is given by a field in the track structure. Each trackpoint has a fixed-length structure as described in the Trackpoint Descriptor Table.

WPT subfile data record

Contains all waypoints.

Offset	Type	Content
6		Waypoint descriptor table info
	uint	Pointer
	uint	Number of table entries
14		Waypoints info
	uint	Pointer
	uint	Number of waypoints

Waypoint descriptor table

The waypoint descriptor table usually follows the fixed fields described above. It defines all the fields that make up one waypoint.

Each entry has the following structure:

Type	Content
ushort	Field type (see below)
ushort	Field length

Field types:

Field type	Length	Datatype of value	Meaning
0	1		?
1	4	MapUnit	Latitude
2	4	MapUnit	Longitude
20	0 if number of tracks =0, else >0	fixstring	Name
21	<i>variable</i>	fixstring	Notes
22	2		Symbol, see table below
23	1	byte	Display mode, see table below
24	4	nfloat	Depth in meter
25	4	nfloat	Temperature in °C
26	4	ndate	TimeStamp

Waypoints

The table of waypoints usually follows the waypoint descriptor table. Each waypoint has a fixed-length structure as described in the Waypoint Descriptor Table.

Enumerations

Color values:

Value	Meaning
0	Black
1	Dark red
2	Dark green
3	Dark yellow
4	Dark blue
5	Dark magenta
6	Dark cyan
7	Light grey
8	Dark grey
9	Red
10	Green
11	Yellow
12	Blue
13	Magenta
14	Cyan
15	White
16	Transparent

Display mode:

Value	Meaning
1	Symbol
3	Symbol & Name
5	Symbol & Comment

Symbol:

Note: There is a similar table in the document "Garmin MapSource (MPS), GPS Database (GDB) and Folder and Item (GFI) File Format", but it is not same! See [4].

Value	Meaning
0	Anchor
1	Bell
2	Diamond, Green
3	Diamond, Red
4	Diver Down Flag 1
5	Diver Down Flag 2
6	Bank
7	Fishing Area
8	Gas Station
9	Horn
10	Residence / House
11	Restaurant
12	Light
13	Bar

Value	Meaning
14	Skull and Crossbones
15	Square, Green
16	Square, Red
17	Buoy, White
18	Waypoint
19	Shipwreck
21	Man Overboard
22	Navaid, Amber
23	Navaid, Black
24	Navaid, Blue
25	Navaid, Green
26	Navaid, Green/Red
27	Navaid, Green/White
28	Navaid, Orange
29	Navaid, Red
30	Navaid, Red/Green
31	Navaid, Red/White
32	Navaid, Violet
33	Navaid, White
34	Navaid, White/Green
35	Navaid, White/Red
36	Dot, White
37	Radio Beacon
150	Boat Ramp
151	Campground
152	Restroom
153	Shower
154	Drinking Water
155	Telephone
156	Medical Facility
157	Information
158	Parking Area
159	Park
160	Picnic Area
161	Scenic Area
162	Skiing Area
163	Swimming Area
164	Dam
165	Controlled Area
166	Danger Area
167	Restricted Area
168	Null
169	Ball Park
170	Car
171	Hunting Area
172	Shopping Center
173	Lodging / Hotel

Value	Meaning
174	Mine
175	Trail Head
176	Truck Stop
177	Exit
178	Flag
179	Circle with X
180	Open 24 Hours
181	Fishing Hot Spot Facility
182	Bottom Conditions
183	Tide/Current Prediction Station
184	Anchor Prohibited
185	Beacon
186	Coast Guard
187	Reef
188	Weed Bed
189	Dropoff
190	Dock
191	Marina / U Marina
192	Bait and Tackle
193	Stump / U Stump
196	Navaid, Dark
197	Exposed Wreck
198	Recommended Anchor
199	Brush Pile
200	Caution
201	Fishing Area 1
202	Fishing Area 2
203	Fishing Area 3
204	Fishing Area 4
205	Fishing Area 5
206	Fishing Area 6
207	Fishing Area 7
208	Fishing Area 8
209	Fishing Area 9
210	Fish Attraktor
211	Hump
212	Laydown
213	Ledge
214	Lilly Pads
215	No Wake Zone
216	Rocks
217	Stop
218	Underwater Grass
219	Underwater Tree
220	Pin, Yellow
221	Flag, Yellow
222	Diamond, Yellow

Value	Meaning
223	Circle, Yellow
224	Square, Yellow
225	Triangle, Yellow
8193	US Hwy
8194	State Hwy
8195	Mile Marker
8196	TracBack Point
8197	Golf Course
8198	City (Small) / Small City
8199	City (Medium)
8200	City (Large)
8201	Intl freeway hwy
8202	Intl national hwy
8203	City (Capitol)
8204	Amusement Park
8205	Bowling
8206	Car Rental
8207	Car Repair
8208	Fast Food
8209	Fitness Center
8210	Movie Theater
8211	Museum
8212	Pharmacy
8213	Pizza
8214	Post Office
8215	RV Park
8216	School
8217	Stadium
8218	Department Store
8219	Zoo
8220	Convenience Store
8221	Live Theater
8222	Ramp intersection
8223	Street intersection
8226	Scales
8227	Toll Booth
8228	Elevation point
8229	Exit without services
8230	Geographic place name, Man-made
8231	Geographic place name, water
8232	Geographic place name, land
8233	Bridge
8234	Building
8235	Cemetery
8236	Church
8237	Civil
8238	Crossing

Value	Meaning
8239	Ghost Town
8240	Levee
8241	Military
8242	Oil Field
8243	Tunnel
8244	Beach
8245	Forest
8246	Summit
8247	Large Ramp intersection
8248	Large exit without services
8249	Police Station
8250	Gambling/casino
8251	Ski Resort
8252	Ice Skating
8253	Wrecker
8254	Border Crossing (Port Of Entry)
8255	Geocache
8256	Geocache Found
8257	Contact, Smiley
8258	Contact, Ball Cap
8259	Contact, Big Ears
8260	Contact, Spike
8261	Contact, Goatee
8262	Contact, Afro
8263	Contact, Deadlocks
8264	Contact, Female1
8265	Contact, Female2
8266	Contact, Female3
8267	Contact, Ranger
8268	Contact, Kung-Fu
8269	Contact, Sumo
8270	Contact, Pirate
8271	Contact, Biker
8272	Contact, Alien
8273	Contact, Bug
8274	Contact, Cat
8275	Contact, Dog
8276	Contact, Pig
8277	Contact, Blonde
8278	Contact, Clown
8279	Contact, Glasses
8280	Contact, Panda
8282	Water Hydrant
8284	Flag, Blue
8285	Flag, Green
8286	Flag, Red
8287	Pin, Blue

Value	Meaning
8288	Pin, Green
8289	Pin, Red
8290	Block, Blue
8291	Block, Green
8292	Block, Red
8293	Bike Trail
8294	Circle, Red
8295	Circle, Green
8296	Circle, Blue
8299	Diamond, Blue
8300	Oval, Red
8301	Oval, Green
8302	Oval, Blue
8303	Rectangle, Red
8304	Rectangle, Green
8305	Rectangle, Blue
8308	Square, Blue
8309	Letter A, Red
8310	Letter B, Red
8311	Letter C, Red
8312	Letter D, Red
8313	Letter A, Green
8314	Letter B, Green
8315	Letter C, Green
8316	Letter D, Green
8317	Letter A, Blue
8318	Letter B, Blue
8319	Letter C, Blue
8320	Letter D, Blue
8321	Number 0, Red
8322	Number 1, Red
8323	Number 2, Red
8324	Number 3, Red
8325	Number 4, Red
8326	Number 5, Red
8327	Number 6, Red
8328	Number 7, Red
8329	Number 8, Red
8330	Number 9, Red
8331	Number 0, Green
8332	Number 1, Green
8333	Number 2, Green
8334	Number 3, Green
8335	Number 4, Green
8336	Number 5, Green
8337	Number 6, Green
8338	Number 7, Green

Value	Meaning
8339	Number 8, Green
8340	Number 9, Green
8341	Number 0, Blue
8342	Number 1, Blue
8343	Number 2, Blue
8344	Number 3, Blue
8345	Number 4, Blue
8346	Number 5, Blue
8347	Number 6, Blue
8348	Number 7, Blue
8349	Number 8, Blue
8350	Number 9, Blue
8351	Triangle, Blue
8352	Triangle, Green
8353	Triangle, Red
8354	Library
8355	Ground Transportation
8356	City Hall
8357	Winery
8359	Asian Food
8360	Deli
8361	Italian Food
8362	Seafood
8363	Steak
8364	ATV
8365	Big Game
8366	Blind
8367	Blood Trail
8368	Cover
8369	Covey
8370	Food Source
8371	Furbearer
8372	Lodge
8373	Small Game
8374	Animal Tracks
8375	Treed Quarry
8376	Tree Stand
8377	Truck
8378	Upland Game
8379	Waterfowl
8380	Water Source
8382	Multi-Cache
8383	Letterbox Cache
8384	Puzzle Cache
16384	Airport
16385	Intersection
16386	Non-directional beacon

Value	Meaning
16387	VHF Omni-range
16388	Heliport
16389	Private Field
16390	Soft Field
16391	Tall Tower
16392	Short Tower
16393	Glider Area
16394	Ultralight Area
16395	Parachute Area
16396	VOR/TACAN
16397	VOR-DME
16398	First approach fix
16399	Localizer Outer Marker
16400	Missed approach point
16401	TACAN
16402	Seaplane Base

References

Used sources of information

Also see the references in the IMG/ADM Container File Format document.

- [1] <https://sourceforge.net/p/adm2gpx/>
- [2] https://wiki.openstreetmap.org/wiki/OSM_Map_On_Garmin/TRK_Subfile_Format
- [3] https://en.wikipedia.org/wiki/Unix_time
- [4] https://github.com/gpsbabel/gpsbabel/blob/master/garmin_tables.cc

Standards and specifications

- [5] ISO/IEC/IEEE 60559:2011 Information technology – Microprocessor Systems – Floating-Point arithmetic (which is the successor of IEEE Std 754-2008)
- [6] 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

Also see the references in the IMG/ADM Container File Format document.

- [7] <https://www.naviboard.de/thread/35720-gpsmap-420-userdata-adm-file-konvertieren/>