

# **DrDepth**

## **File Formats**

Herbert Oppmann

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

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

2021-06-06

## Content

DrDepth File Formats .....	3
Basic data types .....	3
DrDepth Database File Format (*.ddb) .....	3
File structure .....	3
Data Record .....	4
DrDepth Waypoint Notes File Format (*.dnt) .....	4
File structure .....	4
DrDepth Data File Format (*.drd) .....	4
File structure .....	4
Data Record .....	5
DrDepth Mobile Map File Format (*.dmm) .....	6
File structure .....	6
DrDepth Demo File Format (*.ddd) .....	8
DrDepth Waypoint Data File Format (*.dwp) .....	8
References .....	10
Used Sources of information .....	10
Standards and specifications .....	10
Sources of sample files .....	10

## DrDepth File Formats

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

The formats are from the PC program DrDepth, written by Per Pelin [1] , [2]. It was sold to Humminbird (a brand of Johnson Outdoors Inc., [3]) in 2013, and replaced by the program LakeMaster AutoChart.

File formats covered here, sorted by filename extension:

```
*.ddb      Database
*.ddd      Demo
*.dnt      Waypoint notes
*.dmm      Mobile map
*.drd      Data
*.dwp      Waypoint data
```

## 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)
float	4	32 bit floating point value according to [4].
double	8	64 bit floating point value according to [4].
char	1	8-bit character according to probably Windows-1252, which is based on ISO 8859-1.
wchar	2	16-bit character. Only ASCII codes seen, but allowed character set could also be UCS-2 or UTF-16.
rgb	3	Color value. A structure, consisting of 3 byte values for Red, Green and Blue. 0=off, 255=full intensity.

## DrDepth Database File Format (\*.ddb)

### File structure

Type	Content
uint	Probably file type, =1 for DDB or DNT
uint	Number of DRD files
	Repeat number of DRD files
byte	Length of string
char[ <i>len1</i> ]	Full path of DRD file
uint	Number of data records (same as number of DRD files)
	Repeat number of data records
	Data record, see below

## Data Record

Type	Content
double	=10.0
double	Minimum Latitude in °
double	Minimum Longitude in °
double	Maximum Longitude in °
double	Maximum Latitude in °
uint	Numer of data records in the corresponding DRD file
uint	=1

## DrDepth Waypoint Notes File Format (\*.dnt)

Filename drdwptnotes.dwp

### File structure

Type	Content
uint	Probably file type, =1 for DNT or DDB
uint	Number of Waypoint notes
byte	Length of info string, = 78
char[len info]	Contains the following fixed text (CR = Carriage Return, LF = Line Feed): CR + LF + "DrDepth wpt notes" + CR + LF + "!!!!!!!!!!!!!!!!!!!!" + CR + LF + " - Do not edit -" + CR + LF + "!!!!!!!!!!!!!!!!!!!!" + CR + LF + "!"
	Repeat number of Waypoint notes
byte	Length of note string
char[len note]	Name of the waypoint, surrounded by '<' and '>', directly followed by the comment.

## DrDepth Data File Format (\*.drd)

### File structure

Type	Content
uint	Probably file type, =2 for DRD or DMM
byte[3]	=0xFF 0xFE 0xFF
byte	Length of string1 (number of 16-bit characters). Seen value 15.
wchar[len1]	e.g. "Log201210191557" – contains date and time in format YYYYMMDDhhmm
byte[3]	=0xFF 0xFE 0xFF
byte	Length of string2. Is always 14.
wchar[len2]	String. Containing

Type	Content
	a) the software key, e.g. "52-01-71-F6-04" (seen with DrDepth V4.6.5) 5201-F604 is the volume serial number of the first fixed disk (usually C:) of the PC where the program was licensed, and 71 is a fixed value for the standard version of the program. There is also a BT, SS, and BTSS version. - or - b) Probably the license serial number – probably in DrDepth version 5.x 14 hexadecimal digits representing 7 bytes: Seen values "0271EC8337FBC7", "0EA0A2194E1FA8", "1204FF1918665C", "1E17-FA19C3EEAA", "35F9121963765E", "967B5319D73C22", "9DB71719C3EEAA"
double	Some Latitude in ° - note: These are not the minimum - maximum values
double	Some Longitude in °
double	Some Latitude in °
double	Some Longitude in °
uint	Loob value in °
uint	=0
uint	Seen values 0 and 50.
uint	=0
uint	=0
uint	Number of data records
	Repeat number of data records
	Data record, see below

## Data Record

Data records have same size and similar, but not same, layout than with DDB file format.

Type	Content
double	Depth in meter
double	Latitude in °
double	Longitude in °
double	Value1, see below
double	Value2, see below
uint	Seen values 0, 10, 1001-1004 Values 1001-1004 only seen in the first data record, with the following being 10.
uint	TimeOffset in milliseconds. In recorded Data Records, this value must be >= the one from the previous data record. But points inserted later with DrDepth have 0 here.

The two double values are a bit strange. Their content is more dependent on the source of the data, not on the other data fields within this file.

Cases:

- 1) Both Value1 and Value2 are 0.0 → values are unused.
- 2) Value2 == 666.0 → Value1 contains the Water Temperature in Celsius.

3) Both values are in TwoFixed Format (see below)

3a) Value1.FixedVal1 = 222.00, Value2.FixedVal1 = 342.00 and Value2.FixedVal2 = 614.42 → Value1.FixedVal2 = unknown value in range ca. 23-43. This seems to be when the source is Humminbird \*.son + \*.idx files.

3b) Value1.FixedVal1 = 127.00 and Value2.FixedVal2 = 614.43 → Value1.FixedVal2 = unknown value, Value2.FixedVal1 = Water Temperature in Celsius. This seems to be the case when the source is Navico Sonar Log files \*.slg, \*.sl2 or \*.sl3.

### TwoFixed

Two fixed point values are combined and stored as one double floating point value. Each fixed point value has two digits after the decimal point. The allowed value range is 0.00 to 167772.15.

The double floating point value is calculated this way:

Stored value = (FixedVal1 \* 100.0 \* 16777216.0) + (FixedVal2 \* 100.0).

## DrDepth Mobile Map File Format (\*.dmm)

### File structure

Type	Content
uint	Probably file type, =2 for DMM or DRD
uint	Unknown1 = 0
uint	Min depth range in unit below
uint	Max depth range in unit below
uint	Unit for depth range: 0=meter, 1=feet
byte[8]	Software key, e.g. 0x04 0xF6 0x01 0x52 0x71 0x00 0x00 0x00 (seen with DrDepth V4.6.5) 5201-F604 is the volume serial number of the first fixed disk (usually C:) of the PC where the program was licensed, and 0x71 is a fixed value for the standard version of the program. There is also a BT, SS, and BTSS version. For later calculations: SoftwareKey1 are the first four bytes and SoftwareKey2 are the next four bytes of the software key, as uint respectively.
ushort	Unknown2 = 1
ushort	Unknown3 = 2
ushort	Scale factor for depth. 1 = 1 cm, 10 = 10 cm (1 decimeter). The factor 10 is chosen if at least one depth value is more than 32767 cm (327,67 m).
rgb[21]	Display colors
double	Latitude in ° For later calculations: Latitude1 are the first four bytes and Latitude2 are the next four bytes of the latitude, as uint respectively.
double	Longitude in ° For later calculations: Longitude1 are the first four bytes and Longitude2 are the next four bytes of the longitude, as uint respectively.
uint	Some kind of checksum: Check1 = SoftwareKey1 xor Longitude1

Type	Content
uint	Obfuscated left border = left border - (right border - 1)/2
uint	Obfuscated right border = - (right border - 1)/2
uint	Unknown7 = 1
uint	Width
uint	Height
uint	<p>Some kind of checksum:  <math>\text{Check2} = \text{FileType} \text{ xor } \text{Unknown1} \text{ xor } \text{Min depth range} \text{ xor } \text{Max depth range} \text{ xor } \text{Unit} \text{ xor } \text{SoftwareKey1} \text{ xor } \text{SoftwareKey2} \text{ xor } \text{Latitude1} \text{ xor } \text{Latitude2} \text{ xor } \text{Longitude1} \text{ xor } \text{Longitude2} \text{ xor } \text{Check1} \text{ xor } \text{Obfuscated left border} \text{ xor } \text{Obfuscated right border} \text{ xor } \text{Unknown7} \text{ xor } \text{Width} \text{ xor } \text{Height}</math>            For later calculations: Byte1 is the first byte, Byte2 the second byte, Byte3 the third byte and Byte4 the last byte of Check2, as uint respectively.</p>
uint	<p>Obfuscated ByteUK3:            Registry key HKCU\Software\PepeSoft\DrDepthPC\Version 1.0\UserKey contains the license key of the DrDepth installation. It is a string consisting of 10 hexadecimal digits which represent 5 bytes (ByteUK0 to ByteUK4). ByteUK0, ByteUK1 and ByteUK4 are calculated from the Volume Serial Number. They can be determined from the Software key above. ByteUK2 is a checksum across the other four bytes that can be determined once ByteUK3 is known.            This value here enables finding out ByteUK3 by trying out all byte values until the following calculation matches:</p> <p> <math>\text{Byte1} = \text{Byte0} \text{ xor } \text{Byte1}</math>  <math>\text{Byte3} = \text{Byte2} \text{ xor } \text{Byte3}</math>  <math>\text{Byte0} = \text{Byte0} + 4;</math>  <math>\text{Byte2} = \text{Byte2} + 7;</math>            If <math>\text{Byte0} == 0</math>, set Byte0 to 0x91            If <math>\text{Byte1} == 0</math>, set Byte1 to 0x4D            If <math>\text{Byte2} == 0</math>, set Byte2 to 0x52  <math>\text{Byte2} = \text{Byte2} \text{ and } 0x7F</math>            If <math>\text{Byte3} == 0</math>, set Byte3 to 0x42         </p> <p> <math>\text{TmpByte} = \text{ByteUK1} \text{ xor } \text{ByteUK3} \text{ xor } \text{ByteUK4}</math>  <math>\text{Obfuscated ByteUK3} = (\text{Byte0} * 17 + \text{Byte1} + 51) \ll 16 + \text{Byte2} * \text{TmpByte} + \text{Byte3} + 2 * \text{TmpByte}</math> </p>
uint	<p>Obfuscated map creation date:  <math>\text{ulong sys} = \text{GetSystemTimeAsFileTime} (100 \text{ ns intervals since } 1601-01-01)</math>  <math>\text{uint days} = (\text{uint})((\text{sys} + 0xFE624E212AC1B000) / 10000000 / 86400)</math>  <math>+ 0xFE624E212AC1B000</math> is (because of overflow) identical to <math>- 0x19DB1DED53E5000</math>, which is approximately 369 years. Division by 10000000 makes seconds, and the following division by 86400 makes days. This is thus days since 1970-01-01.  <math>\text{uint ticks} = \text{GetTickCount} (\text{milliseconds since the system was started})</math>  <math>\text{Obfuscated map creation date} = ((\text{days} * ((\text{Byte1} \&amp; 0x7F) + 7)) \ll 8) + ((\text{ticks} \gg 5) \&amp; 0xFF)</math>            (Byte1 means the value of Byte1 AFTER calculation of Obfuscated ByteUK3)         </p>
ushort	<p>Orientation            0: Lines represent columns</p>

Type	Content
	1: Lines represent rows
ushort	=0
uint	Number of lines (columns or rows) This is the same value as either Width or Height, and usually the smaller of those two.
	Repeat number of lines
ushort	Starting position within the line, or the maximum length of the line if empty.
ushort	Length of the used part of the line, or 0 if empty.
	Repeat number of non-empty lines
	Repeat length of the used part of the line according to line description above
ushort	Water depth in cm or decimeter, or 0xFFFF for "no value"

The lines run from left to right when number of lines is Height, or from top to bottom when number of lines is Width.

## DrDepth Demo File Format (\*.ddd)

This is just a text file (can be opened in your favorite editor). It contains lines according to NMEA General Sentence Format [5]. Seen talker sentences:

Standard: GPBOD, GPGGA, GPGLL, GPGSA, GPGSV, GPRMB, GPRMC, GPRTE, SDDBT, SDDPT, SDMTW, SDVHW

Garmin proprietary: PGRME, PGRMZ

DrDepth proprietary: TIME

\$TIME is immediately followed by the number of milliseconds since the start of the recording, without comma in between. The first and only data field contains either 'CO' or 'C1' (probably meaning channel 0 or 1).

## DrDepth Waypoint Data File Format (\*.dwp)

This is just a text file (can be opened in your favorite editor). The first line is

```
Drd wp data version 2
```

The last line is

```
End
```

The lines between these take the form

```
<waypoint name>, <latitude>, <longitude>, <depth>, <style>, <icon set>, <icon>
```

The terms in < > have to be replaced by the real values of course.

Term	Type	Meaning
waypoint name	string	Name of the waypoint



Term	Type	Meaning
latitude	double	Latitude in °
longitude	double	Longitude in °
depth	double	Depth in meter
style	uint	0= Icon+Text, 1= Icon only, 2= Text only (centered)
icon set	uint	0= Classic, 1= Lowrance (HDS), 2= Lowrance (LCX), 3= Humminbird, 4= Garmin, 5= User
icon	uint	Index of icon in the icon set

Double: floating point numbers use the dot as decimal separator.

## References

### Used Sources of information

- [1] DrDepth website (archived)  
<https://web.archive.org/web/20121225124133/http://www.drdepth.se:80/>
- [2] DrDepth change history <https://www.fiskesnack.com/forum/fiskeb%C3%A5star-och-fiskekajaker/elektronik/138053-drdepths-nya-%C3%A4gare/page3>
- [3] Humminbird <https://www.humminbird.com/>

### Standards and specifications

- [4] ISO/IEC/IEEE 60559:2011 Information technology – Microprocessor Systems – Floating-Point arithmetic (which is the successor of IEEE Std 754-2008)
- [5] NMEA (National Marine Electronics Association) 0183 standard  
[https://en.wikipedia.org/wiki/NMEA\\_0183](https://en.wikipedia.org/wiki/NMEA_0183)

### Sources of sample files

– none –