

AWE Communications GmbH
Moltkestraße 28a
D-71116 Gärtringen
Germany
Tel.: +49 (0) 7034 / 92 99 52
Fax: +49 (0) 7034 / 92 99 81
e-Mail: mail@winprop.de
WWW: http://www.winprop.de

File Format Description

ASCII and Binary Topographical Data bases

1. Principle Definition of WinProp Topographical Coordinate System

The binary topographical data base file format for the *WinProp* software package is a very simple and efficient data format. The extension of the file is *.tdb (topo data binary). Additionally there is an ASCII file format for the topographical data with the extension *.tda (topo data ASCII).

Both file formats (*.tda and *.tdb) are defined as matrix with constant resolution. The matrix describes a rectangular area. All dimensions (length, width, resolution) are in meters.

Because the topographical areas considered for the prediction models in *WinProp* are not very large, distortions of the rectangular area due to the curved surface of the earth are not taken into account.

If the topographical data is gained from a GIS system the rectangular area must be defined and the values must be transformed to the rectangle. Therefore the type of the coordinate system (UTM-coordinates, Gauss-Krueger,...) is not relevant for the WinProp tools, because the rectangle is independent of the definitions of the coordinate systems.

The height information is stored in meters (e.g. above sea level) in the file. The values for the height must not be absolute values, they can also be relative, because all predictions models do not consider absolute height information.

2. Principle Structure of WinProp Topographical Data Bases

The files (either ASCII or Binary) consist of two blocks:

- Header
General data for the data base.
 - o Coordinates of lower left corner
 - o Resolution
 - o Number of lines and columns in matrix
 - o Factor for height values
- Data (Matrix)
Values for the pixels in the matrix (data base values). Organised

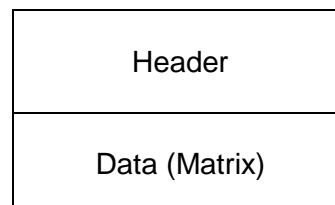


Figure 1: General structure of data base file

The header in ASCII files is neglected. These files consist only of the data matrix. The coordinates of the lower left corner, the number of lines and columns and all further data of the header are determined automatically.

3. Definition of the WinProp ASCII Topographical Data Bases

3.1 WinProp ASCII Topographical Data Base (*.tda)

As already described in section 2, the header in ASCII files is neglected. The files consist only of the data matrix. The coordinates of the lower left corner, the number of lines and columns and all further data of the header are determined automatically.

Each pixel in the topographical data base is defined as a line in the ASCII file. First the x, then the y-coordinate and last the height information. Separation of values either by comma, tabulator or space.

The resolution of the matrix (grid) can be variable, the sequence of pixels in the file is arbitrary.

Definition:

x-coordinate	y-coordinate	height
x-coordinate	y-coordinate	height
x-coordinate	y-coordinate	height

Example:

.....
120.00	200.00	50.00
125.00	200.00	25.00
130.00	200.00	30.00
140.00	200.00	35.00
110.00	200.00	55.00
108.00	205.00	60.00

Predictions are not possible with the ASCII format. The ASCII data bases must be converted to binary data bases. AWE Communications provides a tool for the conversion. During the conversion the user is asked to define the resolution of the new binary file. The lower left corner, the upper right corner and the number of columns and lines are determined automatically. During the conversion all values in the ASCII file are transformed to the new grid of the binary file.

3.2 WinProp ASCII Matrix Topographical Data Base (*.tdm)

Additionally a further version of topographical ASCII data bases are supported with the converter. In this file format the lines and columns of the matrix are also defined as lines and columns in the file. Some header lines define the resolution, the x- and y-coordinates of the lower left corner. Then the values of the matrix come, organised in lines and columns like the real data base. The first value in the first line of data is the upper left corner, the second value in the first line is the value for the pixel next to the first one on the upper boundary of the data base, etc.

These files can also be converted to the binary topographical data format with the converter of AWE Communications.

Example:

LL	10000.00,	20000.00
RESOLUTION	50.0	
250.0	252.0	255.0
248.0	250.0	253.0
246.0	246.0	250.0
247.0	246.0	248.0
...

4. Definition of the WinProp Binary Topographical Data Bases (*.tdb)

4.1 Header

The header contains all relevant data required for the interpretation of the pixel data. Two types of data are used in the header:

- Integer
- Real

They are stored as 4 byte data in the file.

To generate a file format, which is independent of the operating system and microprocessor, the integers and doubles are not saved as integers or doubles. They are converted to a special byte data format and saved/read with dedicated functions (Code for functions available from AWE Communications).

Integer:

4 Byte Integer: B4 B3 B2 B1

Byte 4	$2^{31} \dots 2^{24}$
Byte 3	$2^{23} \dots 2^{16}$
Byte 2	$2^{15} \dots 2^8$
Byte 1	$2^7 \dots 2^0$

Saved in file:

Each byte separately, starting with Byte 1, Byte 2, Byte 3 and last Byte 4.

Example:

Value: 1234 (decimal) = 00 00 04 D2 (Hex)

Saved bytes (Hex values):

D2	04	00	00
----	----	----	----

Negative values are stored the same way, i.e. the sign bit is included in Byte 4.

Real:

Integer part (clipped, not rounded) of the value stored as integer (see above).

Fractional part of the value separated and multiplied with 500,000 and stored as integer (see above).

Example:

Value: 1234.5678

Saved in File:

1234 (decimal as integer) = 00 00 04 D2 (Hex)
(as integer)

Saved bytes:

D2	04	00	00				
----	----	----	----	--	--	--	--

Structure of the header:

Position	Bytes	Data type	Value (Meaning)	Default-Values ¹⁾
01	4	Integer	File Identifier	1852143187
02	4	Integer	Version of file format	1
03	4	Integer	ID (software tool) of generator of file	-1
04	8	Real	Version of software, generated the file	1.0
05	4	Integer	Local time at generation of file (seconds)	0
06	4	Integer	Local time at generation of file (minutes)	0
07	4	Integer	Local time at generation of file (hours)	12
08	4	Integer	Local time at generation of file (day)	1
09	4	Integer	Local time at generation of file (month)	1
10	4	Integer	Local time at generation of file (year)	2000
11	4	Integer	Computation time (with common parts) [s]	0
12	4	Integer	Computation time (without common parts) [s]	0
13	4	Integer	ID of computer (Generator of file)	111
14	4	Integer	ID for Area	0
15	4	Integer	Mode of Data	0
16	4	Integer	Number of lines in file	
17	4	Integer	Number of columns in file	
18	8	Real	Factor of data in matrix ²⁾	
19	8	Real	Resolution of matrix (grid of pixels)	
20	8	Real	x-coordinate of lower left corner	
21	8	Real	y-coordinate of lower left corner	
22	8	Real	z-coordinate of lower left corner	
23	8	Real	x-coordinate of upper right corner	
24	8	Real	y-coordinate of upper right corner	
25	8	Real	z-coordinate of upper right corner	
26	4	Integer	ID for additional bytes (must be 0)	0
27	8	Real	Not used	0.0
28	4	Integer	ID for additional bytes (must be 0)	0

¹⁾ All values are decimal values

²⁾ All values in the matrix are multiplied with this factor and the product is then saved.

4.2 Data

A) Structure of the matrix:

The figure shows the top view on a pixel matrix with five columns and eight lines. The lower left corner (south-west) is pixel 1, the upper right corner (north-east) is pixel 40.

8	16	24	32	40
7	15	23	31	39
6	14	22	30	38
5	13	21	29	37
4	12	20	28	36
3	11	19	27	35
2	10	18	26	34
1	9	17	25	33

The pixels are stored as a sequence of integers in the file, starting with pixel 1 (lower left corner), then 2,3,4,..till the end of the first column. Then the second, third, fourth,... column till the upper right corner of the matrix is reached.

B) Format of a single pixel

Each pixel is stored as integer with 2 bytes (16 bit integer). To achieve a format which is independent of the operating system, the two bytes of each value are stored separately in the file.

Similar to the format of the integers in the header, the following 16 bit data format is specified:

2 Byte Integer: B2 B1

Byte 2	$2^{15} \dots 2^8$
Byte 1	$2^7 \dots 2^0$

Saved in file:

Each byte separately, starting with Byte1.

Example:

Value: 1234 (decimal) = 04 D2 (Hex)

Saved bytes:

D2	04
----	----

Negative numbers are stored the same way, i.e. the sign bit is included in Byte 2.

Saving non-integer values:

The file format is only specified for integer values.

If the values in the matrix are floating-point values with several digits after the point, floating-point values should be stored in the file. To be compatible to the integer file-format, the digits after the point can only be saved if all values in the matrix are multiplied with a common factor (defined in position 18 in the description of the header) and then the results of the multiplication are saved as 16 bit integers. The selection of the factor is arbitrary, but to avoid an overflow (16 bit values!!) and to consider as many digits as possible, it is also very important!