

Tiled To Binary Converter v1.2

Binary File Documentation

1. Data types

Byte	Used mainly for options, can represent a value from 0-255.
Int16	16-bit signed integer (short), it can represent values from -32768 to 32767. This datatype is stored as two bytes with the least significant byte first (little-endian).
UInt16	16-bit unsigned integer (ushort). This is the most common datatype used, can represent values from 0 to 65535. This datatype is stored as two bytes with the least significant byte first (little-endian).
UInt24	24-bit unsigned integer. Used only for RLE encoding (optional), can represent values from 0 to 16,777,215. This datatype is stored as three bytes with the least significant byte first (little-endian).
UInt32	32-bit unsigned integer. Used only for RLE encoding, can represent values from 0 to 16,777,215. This datatype is stored as four bytes with the least significant byte first (little-endian).
String	Strings are ASCII encoded. First a UInt16 is stored that contains string length (l), then the string is saved as a sequence of bytes, one byte per character.

2. Tile Layers

2.1 Header

Orientation	Byte (0=orthogonal, 1=isometric, 2=isometric (staggered), 3=hexagonal (staggered))
Stagger axis	Byte (0=x, 1=y), 0 for orthogonal and isometric.
Stagger index	Byte (0=even, 1=odd), 0 for orthogonal and isometric.
Hex side length	UInt16, will be 0 for other types than hexagonal.

Map width	<i>UInt16</i> , number of tiles horizontally.
Map height	<i>UInt16</i> , (number of tiles vertically).
Tile width	<i>UInt16</i>
Tile height	<i>UInt16</i>
Tile number bits	<i>Byte</i> , (16=16-bit, 32=32-bit). Currently only 16-bit values are saved.
RLE	<i>Byte</i> (0=false, 24=true (24-bit), 32=true (32-bit) whether the layers are stored using run length encoding or not. <i>"Run-length encoding (RLE) is a very simple form of lossless data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run." - Wikipedia</i>
Layer count	<i>UInt16</i> containing the number of layers in the binary file.

2.2 Layers

If RLE is disabled then tile numbers are stored as a sequence of *UInt16*. For RLE the count for a sequence of same tile numbers is stored as a *UInt24* or *UInt32*, then the tile number as *UInt16*. The layers are stored in z-order (background layer comes first).

Tile numbers are stored in this order (table represents a simple map of 5x3 tiles).

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

Tiles are numbered like this (tables represents simple tilesets of 3x3 tiles).

First tileset

0	1	2
3	4	5
6	7	8

Second tileset

9	10	11
12	13	14
15	16	17

3. Object groups

3.1 Header

Object count	<i>UInt16</i> containing the number of objects in the binary file.
Names	<i>Byte</i> , (0=false, 1=true) whether object names are included or not.
Types	<i>Byte</i> , (0=false, 1=true) whether object types are included or not.
Properties	<i>Byte</i> , (0=false, 1=true) whether custom properties are included or not.
Value bits	<i>Byte</i> , (16=16-bit, 32=32-bit). Currently only 16-bit values are saved.

3.2 Object Data

Object data is stored as follows for every object (and in following order):

Object shape	<i>Byte</i> , (0=tile, 1=rectangle, 2=ellipse, 3=polygon, 4=polyline)
Name (optional)	<i>String</i> , if name is missing then only a 0 for string length is saved.
Type (optional)	<i>String</i> , If type is missing then only a 0 for string length is saved.
X-position	<i>UInt16</i>
Y-position	<i>UInt16</i>
Object rotation	<i>Int16</i>
Object number (tile id)	<i>UInt16</i> , 0 if other type than tile.
Object width	<i>UInt16</i> , saved for tile, rectangle, ellipse.
Object height	<i>UInt16</i> , saved for tile, rectangle, ellipse.
Number of points (p)	<i>UInt16</i> , saved for polygons and polylines. Point coordinates below are repeated p times; X1 Y1 X2 Y2 X3 Y3 etc.
Point X	<i>Int16</i> , saved for polygons and polylines.

Point Y	<i>Int16</i> , saved for polygons and polylines.
Number of properties (optional)	<i>UInt16</i> (n) with number of properties. Property name, type and value (below) are repeated n times; name1 type1 value1 name2 type2 value2 etc.
Property name	<i>String</i>
Property type	<i>Byte</i> (0= <i>UInt16</i> , 1= <i>Float</i> , 2= <i>String</i>)
Property value	Stored as <i>UInt16</i> or a <i>String</i> if value is <i>Float</i> or <i>String</i> .

SUMMARIES ON NEXT PAGE! :)

Tile binary file summary

Header

Orientation (Byte)	Stagger axis (Byte)	Stagger index (Byte)	Hex side length (UInt16)	Map width (UInt16)
Map height (UInt16)	Tile width (UInt16)	Tile height (UInt16)	Tile number bits (Byte)	RLE (Byte)
Layer count (UInt16)				

Layers (repeated for every layer)...

Tile number 1 (UInt16)	Tile number 2 (UInt16)	Tile number 3 (UInt16)	Tile number 4 (UInt16)	... until map width * map height
------------------------	------------------------	------------------------	------------------------	----------------------------------

...or layers RLE (repeated for every layer)

Sequence count 1 (3b or 4B)	Sequence tile number 1 (2B)	Sequence count 2 (3b or 4B)	Sequence tile number 2 (2B)	... until map width * map height
-----------------------------	-----------------------------	-----------------------------	-----------------------------	----------------------------------

Object binary file summary

Header

Object count (UInt16)	Names? (Byte)	Types? (Byte)	Properties (Byte)	Value bits (Byte)
-----------------------	---------------	---------------	-------------------	-------------------

Generic object data (repeated for every object)

Object shape (Byte)	Name (optional, string)	Type (optional, string)	X-position (UInt16)	Y-position (UInt16)
Rotation (Int16)	Number (UInt16)			

Specific data for tile, rectangle and ellipse object types (repeated for every valid object)

Width (UInt16)	Height (UInt16)
----------------	-----------------

Specific data for polygons and polylines (repeated for every valid object)

Number of points (UInt16)	Point X1 (Int16)	Point Y1 (Int16)	Point X2 (Int16)	Point Y2 (Int16)
Point X3 (Int16)	Point Y3 (Int16)	... until number of points reached.		

Custom properties (optional, repeated for every object if selected)

Number of properties (UInt16)	Name 1 (String)	Type 1 (Byte)	Value 1 (Int16 or String)	Name 2 (String)
Type 2 (Byte)	Value 2 (Int16 or String)	... until number of properties reached.		