**Tiled Map**

<map>

version:TMX格式版本号，一般为1.0

# orientation:地图朝向，目前支持“正交”和“45度等距”两种方式。

# width:地图的宽度（tile个数isometric则是斜边的tile个数）

# height: 地图的高度（tile个数）

# tilewidth:单Tile的宽度

# tileheight:单Tile的高度

# tilewidth与tileheight两个属性决定了地图网格的大小，个别tileset允许有不同的大小，所有的Tile将靠挂在左下角，超出的部分将在顶部和右侧延伸。

# 可以包含下列元素：属性，图块集，层，对象组

# <tileset>

# **firstgid:**此图块集的第一个图块在全局图块集中的位置。

# 图像来源：如果此图块集来自于一个外部图块定义文件，此属性的值为该图块定义文件路径。TSX文件同样拥有相同的属性结构。在TSX文件中没有firstgid与source属性，如果这个图块集定义文件中的图块在地图中被使用，这两个属性将在TMX文件之中被设置。

# **name:** 图块集的名称

# **tilewidth:** 图块的宽度

# **tileheight:** 图块的高度

# **spacing:** 图块的间距，在原图上采样图块时，图块与图块之间的间隔

# **margin:** The margin around the tiles in this tileset (applies to the tileset image).

# 图块的边距，在原图上采样图块时，图像左侧与上方采样的剔除边界大小

# Can contain: properties (since 0.8.0), image, tile

# 可以包含下列元素：属性（自Tiled0.8.0版本）,图像，图块

# <image>

# format: To be used when embedding images. Deprecated and unsupported in Tiled Qt.

# 嵌入图片时所使用的格式，TiledQT版本当前不支持此属性。

# id: Used by some versions of Tiled Java. Deprecated and unsupported by Tiled Qt.

# 在Tiled Java的某些版本中被使用，TiledQT版本当前不支持此属性。

# **source:** The reference to the tileset image file (Tiled supports most common image formats).

# 图块集对应图像文件的路径。（Tiled支持大多数常见图像格式：bmp,gif,jpeg,jpg,pbm,pgm,png,ppm,tif,tiff,xbm,xpm）

# **trans:** Defines a specific color that is treated as transparent (example value: "FF00FF" for magenta).

# 定义一种用于表示半透的颜色，将原图像中所有相同颜色值的alpha值改为0

# **width:** The image width in pixels (optional, used for tile index correction when the image changes)

# 图像的宽度

# **height:** The image height in pixels (optional)

# 图像的高度

# 当前版本的Tiled QT版本，每个图片集与一张图像对应，按照图片集的属性将该图像切割为多个较小的图块。QT后续开发计划将在一个图片集中支持多个图像，如同JAVA版本的Tiled提供的功能。

# <tile>

# **id:** The local tile ID within its tileset.

# 图块集内部的图块编号

# Can contain: properties, image

# 可以包含下列元素：属性,图像

# <layer>

# **name:** The name of the layer.

# 图层的名称

# x: The x coordinate of the layer in tiles. Defaults to 0 and can no longer be changed in Tiled Qt.

# 该层的X坐标（单位为图块）。默认为0，该值在QT版本的Tiled编辑器中不允许修改。

# y: The y coordinate of the layer in tiles. Defaults to 0 and can no longer be changed in Tiled Qt.

# 该层的Y坐标（单位为图块）。默认为0，该值在QT版本的Tiled编辑器中不允许修改。

# width: The width of the layer in tiles. Traditionally required, but as of Tiled Qt always the same as the map width.

# 该层的宽度（单位为图块）。历史版本要求存在，该值在QT版本的Tiled编辑器中与地图的宽度相同。

# height: The height of the layer in tiles. Traditionally required, but as of Tiled Qt always the same as the map height.

# 该层的高度（单位为图块）。历史版本要求存在，该值在QT版本的Tiled编辑器中与地图的高度相同。

# **opacity:** The opacity of the layer as a value from 0 to 1. Defaults to 1.

# 不透明度，范围0~1,0为全透，1为不透。

# **visible:** Whether the layer is shown (1) or hidden (0). Defaults to 1.

# 是否可见，1可见，0不可见，默认值为1。

# Can contain: properties, data

# 可以包含下列元素：属性,数据块

# <data>

# **encoding:** The encoding used to encode the tile layer data. When used, it can be "base64" and "csv" at the moment.

# 图层数据的编码方式，当前提供两种“base64”和“csv”

# Base64是一种通用的方法，其原理很简单，就是把三个Byte的数据用4个Byte表示。在这四个Byte中，实际用到的都只有前面6bit，这样就不存在只能传输7bit的字符的问题了。Base64的缩写一般是“B”。

# CSV逗号分隔值文件（COMMA Separated value）,是一种用来存储数据的纯文本文件格式。

# **compression:** The compression used to compress the tile layer data. Tiled Qt supports "gzip" and "zlib".

# 图层数据的压缩方式，QT版本的Tiled编辑器支持gzip或者zlib方式的数据压缩。

# When no encoding or compression is given, the tiles are stored as individual XML tile elements. Next to that, the easiest format to parse is the "csv" (comma separated values) format.

# 如果数据为本进行编码转换和压缩的情况下，图块数据将被作为XML的子元素来存储，这是最容易解析的数据格式。

# The base64-encoded and optionally compressed layer data is somewhat more complicated to parse. First you need to base64-decode it, then you may need to decompress it. Now you have an array of bytes, which should be interpreted as an array of unsigned 32-bit integers using little-endian byte ordering.

# 采用base64编码格式的数据和压缩过后的数据解析起来较为复杂。首先你必须对数据进行解码，然后你可能需要加压缩数据。你的数据存储于一个字节数组，你不得不将它排序为一个little-endian字节序的无符号整形数组。

# Whatever format you choose for your layer data, you will always end up with so called "global tile IDs" (gids). They are global, since they may refer to a tile from any of the tilesets used by the map. In order to find out from which tileset the tile is you need to find the tileset with the highest firstgid that is still lower or equal than the gid. The tilesets are always stored with increasing firstgids.

# 无论采用哪种格式存储你的图层数据，你都必须以所谓的gids作为结尾。它们是全局的，自从图块集的任意图块被用于设计当前地图。为了能够找到图块所在的图块集，你需要找到小于图块gid的最大的图块集的firstgid。所有图块集的firstgids都是按顺序排列的。

# When you use the tile flipping feature added in Tiled Qt 0.7.0, the highest two bits of the gid store the flipped state. Bit 32 is used for storing whether the tile is horizontally flipped and bit 31 is used for the vertically flipped tiles. And since Tiled Qt 0.8.0, bit 30 means whether the tile is flipped (anti) diagonally, enabling tile rotation. These bits have to be read and cleared before you can find out which tileset a tile belongs to.

# 当你使用QT版本的Tiled编辑0.7.0提供的图块翻转特性时，gid的高两位存储的是当前图块的翻转状态。第32位用来存储这个图块是否被水平翻转，第31位用来存储这个图块是否被垂直翻转。

# <tile>

# **gid:** The global tile ID.

# 图块的全局ID

# Not to be confused with the tile element inside a tileset, this element defines the value of a single tile on a tile layer. This is however the most inefficient way of storing the tile layer data, and should generally be avoided.

# 不用因为图块存在于图块集中而感到困惑，这个元素定义了可用于图层的一个基本图块。这是最有效率的层数据存储方式，通常应该避免。

# <objectgroup>

# **name:** The name of the object group.

# 对象组的名称。

# **color:** The color used to display the objects in this group.

# 当前对象组的颜色。

# x: The x coordinate of the object group in tiles. Defaults to 0 and can no longer be changed in Tiled Qt.

# 对象组的X坐标。默认为0，TiledQT版本此属性无法修改。

# y: The y coordinate of the object group in tiles. Defaults to 0 and can no longer be changed in Tiled Qt.

# 对象组的Y坐标。默认为0，TiledQT版本此属性无法修改。

# width: The width of the object group in tiles. Meaningless.

# 对象组的宽度，当前无意义。

# height: The height of the object group in tiles. Meaningless.

# 对象组的高度，当前无意义。

# **opacity:** The opacity of the layer as a value from 0 to 1. Defaults to 1.

# 对象组的图透明度0~1，默认为1不透。

# **visible:** Whether the layer is shown (1) or hidden (0). Defaults to 1.

# 对象组的可见性 1显示 0隐藏，默认为1。

# The object group is in fact a map layer, and is hence called "object layer" in Tiled Qt.

# 对象组事实上就是一个图层，因此在TiledQt版本中我们也称之为对象层。

# Can contain: properties, object

# 可以包含下列元素：属性，对象

# <object>

# **name:** The name of the object. An arbitrary string.

# 对象名称。字符串表示

# **type:** The type of the object. An arbitrary string.

# 对象类型。字符串表示

# **x:** The x coordinate of the object in pixels.

# 对象的X坐标，单位像素

# **y:** The y coordinate of the object in pixels.

# 对象的Y坐标，单位像素

# **width:** The width of the object in pixels.

# 对象的宽度，单位像素

# **height:** The height of the object in pixels.

# 对象的高度，单位像素

# **gid:** An reference to a tile (optional).

# 图块的索引（可选）

# **visible:** Whether the object is shown (1) or hidden (0). Defaults to 1.

# 可见性，1显示0隐藏，默认为1

# While tile layers are very suitable for anything repetitive aligned to the tile grid, sometimes you want to annotate your map with other information, not necessarily aligned to the grid. Hence the objects have their coordinates and size in pixels, but you can still easily align that to the grid when you want to.

# 虽然块状图层非常适合用来把所有的物件对齐到网格，但有时你需要在地图中添加一些不需要对齐到网格的信息。因此对象的位置、大小单位为像素，如果你想要对齐对象到网格也很容易。

# You generally use objects to add custom information to your tile map, such as spawn points, warps, exits, etc.

# 通常你用对象来添加一些自定义信息，例如出生点，跳转点，退出点等等。

# When the object has a gid set, then it is represented by the image of the tile with that global ID. Currently that means width and height are ignored for such objects. The image alignment currently depends on the map orientation. In orthogonal orientation it's aligned to the bottom-left while in isometric it's aligned to the bottom-center.

# 如果某个对象的gid属性被设置，对象将用该ID对应的图块来表示，也就是说该对象的宽度和高度信息是被忽略的。图片的对齐方式取决于当前地图的朝向。在正交地图中，于左下角对齐，在等距地图中于底部中心对齐。

# Can contain: properties, polygon, polyline, image

# 可以包含下列元素：属性，多边形，多段线段，图片

# <polygon>

# **points:** A list of x,y coordinates in pixels.

# 一系列X,Z坐标点（单位像素）

# Each polygon object is made up of a space-delimited list of x,y coordinates. The origin for these coordinates is the location of the parent object. By default, the first point is created as 0,0 denoting that the point will originate exactly where the object is placed.

# 多边形对象由一系列空格符分隔的坐标点组成。这些坐标点的原点是父对象的位置。默认情况下，第一个点的坐标为0,0 指示对象的存放位置。

# <polyline>

# **points:** A list of x,y coordinates in pixels.

# 一系列X,Z坐标点（单位像素）

# A polyline follows the same placement definition as a polygon object.

# 规则如同多边形。

# <properties>

# Can contain: property

# 可以包含下列元素：属性

# Wraps any number of custom properties. Can be used as a child of the map, tile (when part of a tileset), layer, objectgroup and object elements.

# 可以包含任意数量的自定义属性。可以用来作为地图，图块，层，对象组，对象的子元素。

# <property>

# **name:** The name of the property.

# 属性名

# **value:** The value of the property.

# 属性值

# When the property spans contains newlines, the current versions of Tiled Java and Tiled Qt will write out the value as characters contained inside theproperty element rather than as the value attribute. However, it is at the moment not really possible to edit properties consisting of multiple lines with Tiled.

# 如果属性值中包含换行符，当前JAVA和QT版本的Tiled编辑器的处理方法是将其包含在property元素中而非value值，当前Tiled编辑器并没有真正意义上支持多行属性值。

# It is possible that a future version of the TMX format will switch to always saving property values inside the element rather than as an attribute.

# 也许在将在的版本中属性值将直接存储于元素中而非作为一个attribute存储。