

Unofficial Half-Life WAD3 and SPRITE file format specification

WAD3

File structure:

- WAD header
- Textures array (image, palette, mipmaps)
- Lumps list (contains basic informations about textures)

WAD header		
Bytes	Type	Description
4	char	File ID. Every HL 1 WAD file starts with string: "WAD3" (0x57, 0x41, 0x44, 0x33)
4	uint	Number of all textures
4	uint	Offset of lumps list in WAD file

Lump item info		
Bytes	Type	Description
4	uint	Offset of texture in WAD file
4	uint	Compressed length of texture
4	uint	Full length of texture
1	byte	Type of texture - regularly 0x43, but there also some other formats: 0x40 (tempdecals.wad...), 0x42 (cached.wad...), 0x46 (format for fonts, gfx.wad...).
1	byte	Compression type (0 = none)
2	byte	Padding*
16	char	Texture name (should be null-padded) , it can start with some special symbol: { = transparent ! = water + = toggling - = random tiling ~ = something like computers, lights

Texture (types: 0x40, 0x42, 0x43, 0x46)			
Bytes	Texture types	Type	Description
16	0x40, 0x43	char	Texture name (should be null-padded)
4	all	uint	Texture width (If type is 0x46 width is often 256)
4	all	uint	Texture height
4	0x46 (font)	uint	Row count
4	0x46 (font)	uint	Row height
256*4	0x46 (font)	uint[2]	[0] = Start offset, [1] = Char width
4	0x40, 0x43	uint	Offset of image from Texture start
4	0x40, 0x43	uint	Offset of mipmap #1 from Texture start

4	0x40, 0x43	uint	Offset of mipmap #2 from Texture start
4	0x40, 0x43	uint	Offset of mipmap #3 from Texture start
n	all	byte	Image data contains bytes indexing it to the palette $n = \text{texture width} * \text{texture height}$
n	0x40, 0x43	byte	Mipmap data 1. $n = (\text{texture width} / 2) * (\text{texture height} / 2)$
n	0x40, 0x43	byte	Mipmap data 2. $n = (\text{texture width} / 4) * (\text{texture height} / 4)$
n	0x40, 0x43	byte	Mipmap data 3. $n = (\text{texture width} / 8) * (\text{texture height} / 8)$
2	all	byte	Unknown 2-bytes (always): 0x00, 0x01
256*3	0x42, 0x43	byte	List of RGB 24bit colors that are 256 entries long. RGB = 0,0,255 means transparent color in HL engine.
2	all	byte	Padding*

SPRITE

File structure:

- Sprite header
- Color palette
- Frames array

Sprite header		
Bytes	Type	Description
4	char	File ID. Every HL 1 SPR file starts with string: "IDSP"(0x49, 0x44, 0x53, 0x50)
4	uint	Sprite version (should be version 2)
4	uint	Sprite type: 0 = VP_PARALLEL_UPRIGHT, 1 = FACING_UPRIGHT, 2 = VP_PARALLEL, 3 = ORIENTED, 4 = VP_PARALLEL_ORIENTED
4	uint	Texture format: 0 = SPR_NORMAL 1 = SPR_ADDITIVE 2 = SPR_INDEXALPHA 3 = SPR_ALPHATEST
4	float	Bounding radius: $\text{sqrt}((\text{Max.width} \gg 1) * (\text{Max.width} \gg 1) + (\text{Max.height} \gg 1) * (\text{Max.height} \gg 1))$
4	uint	Maximum width of frame
4	uint	Maximum height of frame
4	uint	Number of frames

4	float	<i>Beam length</i>
4	uint	<i>Synchronization type (0= synchronized, 1= random)</i>

Color palette		
Bytes	Type	Description
2	uint16	<i>Size of palette</i>
n	byte	<i>List of RGB 24bit colors. n = size of palette * 3</i>

Frame		
Bytes	Type	Description
4	uint	<i>Frame group</i>
4	int	<i>Frame origin X</i>
4	int	<i>Frame origin Y</i>
4	uint	<i>Frame width</i>
4	uint	<i>Frame height</i>
n	byte	<i>Image data contains bytes indexing it to the palette. n = frame width * frame height</i>

Notes:

- Padding* = usually null bytes, not used.
- All textures must be in dimensions that are multiples of 16 and the total size must be less than 10752!

Revision: 05
Author: Yuraj
Date: 15.01.2012
<http://yuraj.ucoz.com/>