Bitmap Font Generator - Documentation

Back to main page

File format

File tags

info

This tag holds information on how the font was generated.

face	This is the name of the true type font.
size	The size of the true type font.
bold	The font is bold.
italic	The font is italic.
charset	The name of the OEM charset used (when not unicode).
unicode	Set to 1 if it is the unicode charset.
stretchH	The font height stretch in percentage. 100% means no stretch.
smooth	Set to 1 if smoothing was turned on.
aa	The supersampling level used. 1 means no supersampling was used.
padding	The padding for each character (up, right, down, left).
spacing	The spacing for each character (horizontal, vertical).
outline	The outline thickness for the characters.

common

This tag holds information common to all characters.

lineHeight	This is the distance in pixels between each line of text.
base	The number of pixels from the absolute top of the line to the base of the characters.
scaleW	The width of the texture, normally used to scale the x pos of the character image.
scaleH	The height of the texture, normally used to scale the y pos of the character image.
pages	The number of texture pages included in the font.
packed	Set to 1 if the monochrome characters have been packed into each of the texture channels. In this case alphaChnl describes what is stored in each channel.
alphaChnl	Set to 0 if the channel holds the glyph data, 1 if it holds the outline, 2 if it holds the glyph and the outline, 3 if its set to zero, and 4 if its set to one.
redChnl	Set to 0 if the channel holds the glyph data, 1 if it holds the outline, 2 if it holds the glyph and the outline, 3 if its set to zero, and 4 if its set to one.
greenChnl	Set to 0 if the channel holds the glyph data, 1 if it holds the outline, 2 if it holds the glyph and the outline, 3 if its set to zero, and 4 if its set to one.
blueChnl	Set to 0 if the channel holds the glyph data, 1 if it holds the outline, 2 if it holds the glyph and the outline, 3 if its set to zero, and 4 if its set to one.

page

This tag gives the name of a texture file. There is one for each page in the font.

id	The page id.
file	The texture file name.

char

This tag describes on character in the font. There is one for each included character in the font.

id	The character id.
x	The left position of the character image in the texture.
y	The top position of the character image in the texture.
width	The width of the character image in the texture.
height	The height of the character image in the texture.
xoffset	How much the current position should be offset when copying the image from the texture to the screen.
yoffset	How much the current position should be offset when copying the image from the texture to the screen.
xadvance	How much the current position should be advanced after drawing the character.
page	The texture page where the character image is found.
chnl	The texture channel where the character image is found (1 = blue, 2 = green, 4 = red, 8 = alpha, 15 = all channels).

kerning

The kerning information is used to adjust the distance between certain characters, e.g. some characters should be placed closer to each other than others.

first	The first character id.
second	The second character id.
amount	How much the x position should be adjusted when drawing the second character immediately following the first.

Binary file layout

This section describes the layout of the tags in the binary file format. To understand what each tag means refer to the file tags section.

The first three bytes are the file identifier and must always be 66, 77, 70, or "BMF". The fourth byte gives the format version, currently it must be 3.

- Version 1 (introduced with application version 1.8).
- Version 2 (introduced with application version 1.9) added the outline field in the infoBlock and the encoded field in the commonBlock.
- Version 3 (introduced with application version 1.10) removed the encoded field in the commonBlock, and added the alphaChnl, redChnl, greenChnl, blueChnl instead. The size of each block is now stored without accounting for the size field itself. The character id in the charsBlock and the kerningPairsBlock was increased to 4 bytes to support the full unicode range.

Following the first four bytes is a series of blocks with information. Each block starts with a one byte block type identifier, followed by a 4 byte integer that gives the size of the block, not including the block type identifier and the size value.

Block type 1: info

field	size	type	pos	comment
fontSize	2	int	0	
bitField	1	bits	2	bit 0: smooth, bit 1: unicode, bit 2: italic, bit 3: bold, bit 4: fixedHeigth, bits 5-7: reserved
charSet	1	uint	3	
stretchH	2	uint	4	
aa	1	uint	6	
paddingUp	1	uint	7	
paddingRight	1	uint	8	
paddingDown	1	uint	9	
paddingLeft	1	uint	10	
spacingHoriz	1	uint	11	
spacingVert	1	uint	12	
outline	1	uint	13	added with version 2
fontName	*n*+1	string	14	null terminated string with length *n*

This structure gives the layout of the fields. Remember that there should be no padding between members. Allocate the size of the block using the blockSize, as following the block comes the font name, including the terminating null char. Most of the time this block can simply be ignored.

Block type 2: common

field	size	type	pos	comment
lineHeight	2	uint	0	
base	2	uint	2	
scaleW	2	uint	4	
scaleH	2	uint	6	
pages	2	uint	8	
bitField	1	bits	10	bits 0-6: reserved, bit 7: packed
alphaChnl	1	uint	11	
redChnl	1	uint	12	
greenChnl	1	uint	13	
blueChnl	1	uint	14	

Block type 3: pages

field	size	type	pos	comment
pageNames	*p**(*n*+1)	strings	0	*p* null terminated strings, each with length *n*

This block gives the name of each texture file with the image data for the characters. The string pageNames holds the names separated and terminated by null chars. Each filename has the same length, so once you know the size of the first name, you can easily determine the position of each of the names. The id of each page is the zero-based index of the string name.

Block type 4: chars

field	size	type	pos	comment
id	4	uint	0+*c**20	These fields are repeated until all characters have been described
x	2	uint	4+*c**20	
y	2	uint	6+*c**20	
width	2	uint	8+*c**20	
height	2	uint	10+*c**20	
xoffset	2	int	12+*c**20	
yoffset	2	int	14+*c**20	
xadvance	2	int	16+*c**20	
page	1	uint	18+*c**20	
chnl	1	uint	19+*c**20	

The number of characters in the file can be computed by taking the size of the block and dividing with the size of the charInfo structure, i.e.: numChars = charsBlock.blockSize/20.

Block type 5: kerning pairs

field	size	type	pos	comment
first	4	uint	0+*c**10	These fields are repeated until all kerning pairs have been described
second	4	uint	4+*c**10	
amount	2	int	8+*c**6	

This block is only in the file if there are any kerning pairs with amount differing from 0.