

The 8bit JControl Font Definition Format (JCFD)

	= <header><fontdata>
<header>	= "JCFD"<revision><size><extcharoffset>
<revision>	= 0x02 (16bit)
<size>	= font height in pixels (16bit)
<extcharoffset>	= offset from start of ASCII-Chars to start of extended chars (increment from "current" address, not from start of file) (16bit)
<fontdata>	= <asciichars><extchars><termination>
<asciichars>	= <binary_tree> including 128 standard 7bit-ascii chars
<extchars>	= { <unicode><chardata> } +
<termination>	= 0x0000 (16bit) indicates the end of the font data
<binary_tree>	= <rightsubtree_offset><left_subtree><right_subtree>
<rightsubtree_offset>	= index (position) of the root of the right subtree
<left_subtree>	= if the height of the tree is smaller than 6 <binary_tree> else <chardata>
<right_subtree>	= if the height of the tree is smaller than 6 <binary_tree> else <chardata>
<unicode>	= the unicode for this character (16bit)
<chardata>	= <width><pixeldata>
<width>	= the character width in pixels (8bit)
<pixeldata>	= { <byte_value_row> } +
<byte_value_row>	= { <byte_value> } * width
<byte_value>	= 8 pixels are stored in a byte, one upon the other, LSB first (8bit)

Thus, image `pixeldata` is always stored as a multiple of eight***<width>**.

Example: font size = 9, character width = 7

								
								..XXX..	
								.X...X.	
								XX...XX	
								XX...XX	
								XXXXXXXX	
								XX...XX	
								XX...XX	
								XX...XX	
LSB		
	.	.	X	X	X	.	.		
	.	X	.	.	.	X	.		
	X	X	.	.	.	X	X		
	X	X	.	.	.	X	X		
	X	X	X	X	X	X	X		
	X	X	.	.	.	X	X		
MSB	X	X	.	.	.	X	X		
	F8	FC	22	22	22	FC	F8		
LSB	X	X	.	.	.	X	X		TH
	:	:	:	:	:	:	:		
MSB		F8
	01	01	00	00	00	01	01		

The pixel data for this character would be stored this way:

F8 FC 22 22 22 FC F8 01 01 00 00 00 01 01

The 16-Bit JControl Font Definition Format (JCFD)

**** = **<header><lookuptable><fontdata>**

<header> = **"JCFD"<revision><size><reserved><offset><chars_num><special_chars_num>**

<revision> = 0x01 (16-Bit)

<size> = font height in pixels (16-Bit)

<reserved> = 0x00 (16-Bit)

<offset> = 0x08 (16-Bit)

<chars_num> = 0x80 (16-Bit) number of normal characters, this should always be 128

<special_chars_num> = number of special chars that come after 0x80, like ä,ö,ü. (16-Bit)

The lookup table includes the width and the start index for every character in this font.

<lookuptable> = **<chars>[<special_chars>]**

<chars> = { **<width><index>** }+ (**<chars_num>** times)

<special_chars> = { **<unicode><width><index>** }+ (**<special_chars_num>** times)

<width> = the character width in pixels (16-Bit)

<index> = the start index for the characters pixel data (16 -Bit)

<unicode> = the unicode for this character (16-Bit)

<fontdata> = { **<pixel_data>** }+

<pixel_data> = { **<letter_column>** }+ (**<width>** times)

<letter_column> = { **<short_value>** }+ (if **<size>** is more than 16, more shorts are needed)

<short_value> = pixels are stored in a short, one upon the other, LSB first (16-Bit each)

Example : font size = 20, character width = 13

```
. . . . .
. . . . .
. . . . .
. . . . XXX . . . .
. . . . X . X . . . .
. . . . XX . XX . . . .
. . . . XX . XX . . . .
. . . . X . . . X . . . .
. . . XX . . . XX . . .
. . . XX . . . XX . . .
. . XX . . . . XX . .
. . XX . . . . XX . .
. XXXXXXXXXXXX .
. XX . . . . . XX .
. XX . . . . . XX .
XX . . . . . XX
XX . . . . . XX
. . . . .
. . . . .
```

LSB

	X	X	X
	X	.	X
	X	X	.	X	X
	X	X	.	X	X
	X	.	.	.	X
	.	.	.	X	X	.	.	.	X	X	.	.	.
	.	.	X	X	X	X	.	.	.
	.	.	X	X	X	X	.	.	.
	.	X	X	X	X	X	X	X	X	X	X	X	.
	.	X	X	X	X	.	.
	.	X	X	X	X	.	.
MSB	X	X	X	X	
	8000	f000	7c00	1f00	13e0	1078	1008	1078	13e0	1f00	7c00	f000	8000
LSB	X	X	X	X

	0001	0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	0001

The pixel data for this character would be stored this way:

```

80 00 00 01 f0 00 00 01 7c 00 00 00 1f 00 00 00
13 e0 00 00 10 78 00 00 10 08 00 00 10 78 00 00
13 e0 00 00 1f 00 00 00 7c 00 00 00 f0 00 00 01
80 00 00 01

```