Font File Format

Publication date: December 2003

Sony Computer Entertainment Inc.
1-1, Akasaka 7-chome, Minato-ku
Tokyo 107-0052, Japan

Sony Computer Entertainment America
919 E. Hillsdale Blvd.
Foster City, CA 94404, U.S.A.

Sony Computer Entertainment Europe
30 Golden Square
London W1F 9LD, U.K.

Table of Contents

About This Manual

This manual is the PS2 Programmer Tool Runtime Library libpfont, Version 1.2 version of the *Font File Format* manual.

Changes Since Last Release

- None

Related Documentation

Refer also to the *Font File Format* manual and the *Simple Font Library Overview* manual.

Note: the Developer Support Web site posts current developments regarding the Libraries and also provides notice of future documentation releases and upgrades.

Typographic Conventions

Certain Typographic Conventions are used throughout this manual to clarify the meaning of the text:

Convention	Meaning
`courier`	Indicates literal program code.
italic	Indicates names of arguments and structure members (in structure/function definitions only).
medium bold	Indicates data types and structure/function names (in structure/function definitions only).
blue	Indicates a hyperlink.

Developer Support

Sony Computer Entertainment America (SCEA)

SCEA developer support is available to licensees in North America only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
Attn: Developer Tools Coordinator Sony Computer Entertainment America 919 East Hillsdale Blvd. Foster City, CA 94404, U.S.A. Tel: (650) 655-8000	E-mail: scea_support@ps2-pro.com Web: https://www.ps2-pro.com/ Developer Support Hotline:(650) 655-5566 (Call Monday through Friday, 8 a.m. to 5 p.m., PST/PDT)

Sony Computer Entertainment Europe (SCEE)

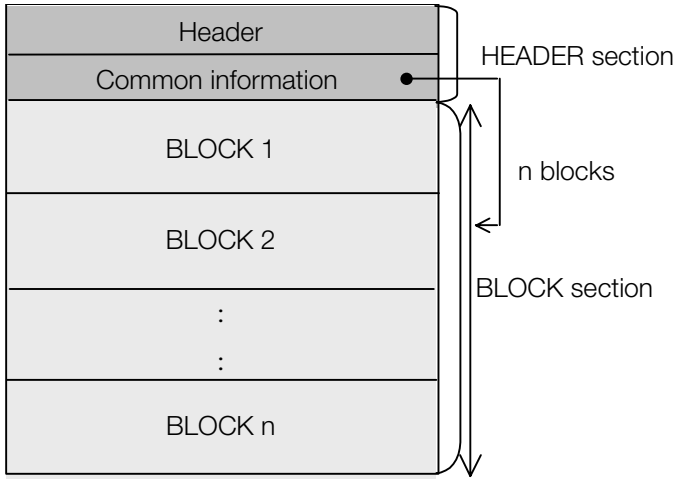
SCEE developer support is available to licensees only in the PAL television territories (including Europe and Australasia). You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
Attn: Development Tools Manager Sony Computer Entertainment Europe 30 Golden Square London W1F 9LD, U.K. Tel: +44 (0) 20 7859-5000	E-mail: scee_support@ps2-pro.com Web: https://www.ps2-pro.com/ Developer Support Hotline: +44 (0) 20 7859-5777 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT/BST)

File Format

Overall Structure

Figure 1



Except where otherwise indicated, numeric values are always treated as signed.

Reserved areas and padding areas are always filled with zeroes.

HEADER Section

Header

Table 1

Name	Bytes
ID	4 (0x00000000:unsigned)
VERSION	4 (0x00000000:unsigned)
SIZE	4 (>0:byte count (includes ID))
RESERVED	4 (0)

Common Information

Table 2

Name	Bytes
Font name	31+1 (UTF8'¥0')
Comment	31+1 (UTF8'¥0')
Overall maximum font ASCENT	2 (>=0: Calculated after applying a scale to each block)
Overall maximum font DESCENT	2 (<=0: Calculated after applying a scale to each block)
Overall maximum font WIDTH	2 (>0: Calculated after applying a scale to each block)
reserved	2 (0)
BLOCK count	4 (>=0:n)

Name	Bytes
BLOCK 1 offset	4 (>0: from start of header)
BLOCK 2 offset	4 (>0: from start of header)
:	:
BLOCK n offset	4 (>0: from start of header)

BLOCK Section Structure

Figure 2

BLOCK header
BLOCK header offset table
image
codeindex
codemap
proportional
kerning
clut

BLOCK Header Information

BLOCK must start on a qword boundary.

Table 3

Name	Bytes	
ID	4 (0x00000000:unsigned)	
VERSION	4 (0x00000000:unsigned)	
SIZE	4 (>0:byte count (includes ID))	
RESERVED	4 (0)	
Flag	4 ()	
	bits 0-2	3-bit TEXTURE color mode 000…4bit index 32 (8H x 4V pixels for image is recommended) 001…8bit index 64 (8H x 2V pixels for image is recommended) 010…16bit direct 64 (4H x 2V pixels for image is recommended) 011…24bit direct 192 (8H x 2V pixels for image is recommended) 100…32bit direct 64 (2H x 2V pixels for image is recommended)
	bit 3	1-bit proportional format 0…fixed 1…individual
	bits 4-31	28 bits Reserved(0)

Name	Bytes
Output size corrected value (X)	4 (float)
Output size corrected value (Y)	4 (float)
Image 1 character width (pixels)	2 (>0)
Image 1 character height (pixels)	2 (>0)
Max ASCENT in BLOCK	2 (>=0)
Max DESCENT in BLOCK	2 (<=0)
Max WIDTH in BLOCK	2 (>0)
reserved	2 (0)

BLOCK Header (offset table)

Table 4

Name	Bytes
Character count	4 (>0:M)
Offset to start of image data	4 (>0: from start of BLOCK, on qword boundary, in bytes)
codeindex count	4 (>0:N)
Offset to start of codeindex data	4 (>=0: from start of BLOCK, on qword boundary, in bytes)
codemap count	4 (>0:L)
Offset to start of codemap data	4 (>=0: from start of BLOCK, on qword boundary, in bytes)
proportional count	4 (>0 or 1:M)
Offset to start of proportional data	4 (>=0: from start of BLOCK, on qword boundary, in bytes)
kerning count	4 (>=0:K Currently fixed at 0)
Offset to start of kerning data	4 (>=0: from start of BLOCK, on qword boundary, in bytes)
clut entry count	4 (>=0)
Offset to start of clut data	4 (>=0)

Font Data Information

Image Information

Images are always in 128-bit units. size(qword)=(bitcount*w*h+127)/128

Insert zeroes for padding.

Figure 3

Character 0

Name	Byte count
IMAGE DATA	X

Character 1

…

Character M-1

codeindex Information

The ranges of neighboring codeindexes must not overlap.

codeindexes must be sorted in ascending order.

A start character must always be present, specified by character code.

Figure 4

Number 0

Name	Byte count
Starting character code	4 (UCS2)
Ending character code	4 (UCS2)
Starting map number	4 (0~L-1)
Starting character number	4 (0~M-1)

Number 1

…

Number N-1

codemap Information

The codemap indexes must be sorted in ascending order.

If 0xffffU appears in the index, the code it references is invalid (i.e. no character).

Figure 5

Number 0

Name	Byte count
index	2 (0~65534 or 65535:unsigned)

Number 1

…

Number L-1

proportional Information

When fixed, proportional information must be set to 1. When individual, information for M characters must be present.

Figure 6

Character 0

Name	Byte count
BASE POINT X	2 (>=0)
BASE POINT Y	2 (>=0)
L BEARING	2 (<=0)
R BEARING	2 (>=0)
ASCENT	2 (>=0)
DESCENT	2 (<=0)
WIDTH	2 (>0)
kerningTAG	2 (0:reserved)

Character 1

…

Character M-1

kerning Information

Reserved (currently fixed at 0)

clut Information

Data for GS PSMCT32-format data clut entries is arranged in ascending order by entry number.

Data Contents

How to Request a Character Number within a Specific Code Block

```
int search(int code){
  for(n = 0; n < N; n++){
    if((code >= codeindex[n].starting character code) && (code <=
codeindex[n].ending character code)){
      int ofs = code - codeindex[n].starting character code;
      u_char idx = codemap[codeindex[n].starting map number + ofs];
      if((0 != ofs) && (0 == idx)){
        return -1; // Specified character not in this block
      }
      return codeindex[n].starting character number + idx; // Found character
number
    }
  }
  return -1; // Specified character not in this block
}
```

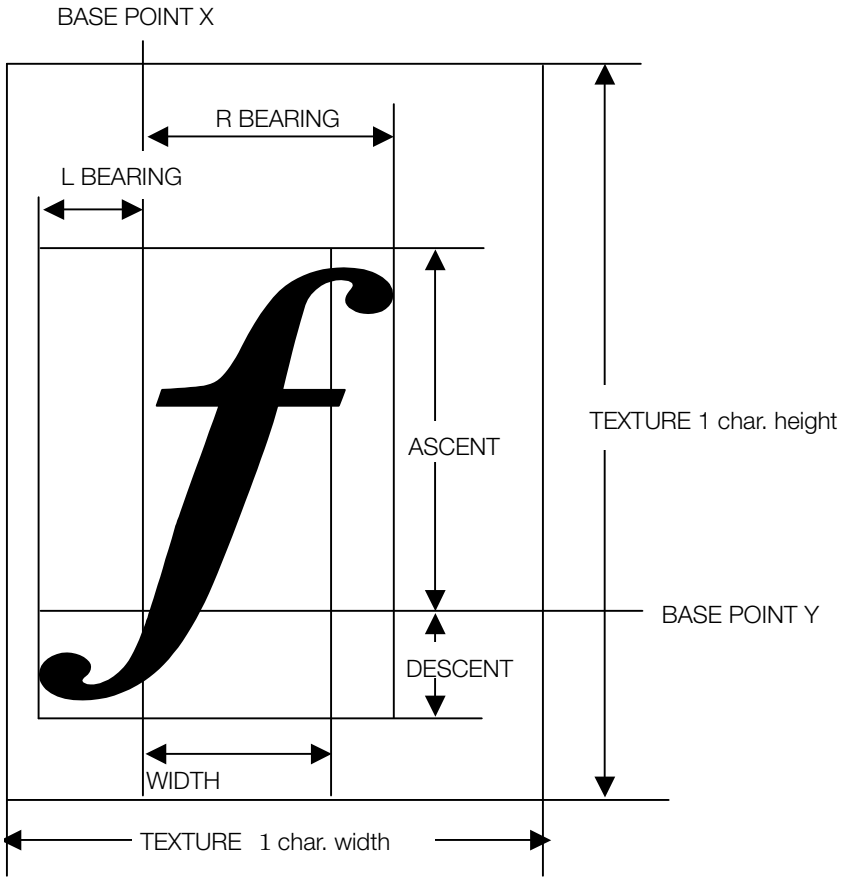
codemap delimits the characters actually found within the range specified by codeindex.

Priority of Codes

If the same character code is present in multiple codemaps or multiple blocks, priority is given in order of appearance.

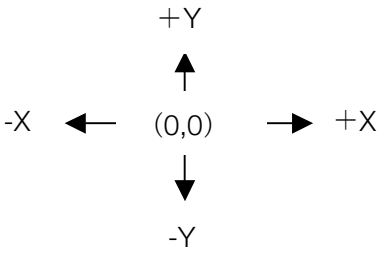
Proportional Data

Figure 7



Codes for ASCENT, DESCENT, R BEARING, and L BEARING are all calculated from a base point (X,Y) at the origin (0,0) as shown below.

Figure 8



As such, ASCENT and R BEARING have positive values, DESCENT and L BEARING have negative values. All other values are positive. Kerning information tags are reserved.