



Linux Ubuntu 22.4.5 Manual Pages on command 'Font::TTF::Utils.3pm'

\$ man Font::TTF::Utils.3pm

Font::TTF::Utils(3pm) User Contributed Perl Documentation Font::TTF::Utils(3pm)

NAME

Font::TTF::Utils - Utility functions to save fingers

DESCRIPTION

Lots of useful functions to save my fingers, especially for trivial tables

FUNCTIONS

The following functions are exported

`($val, $pos) = TTF_Init_Fields ($str, $pos)`

Given a field description from the "DATA" section, creates an absolute entry in the fields associative array for the class

`TTF_Read_Fields($obj, $dat, $fields)`

Given a block of data large enough to account for all the fields in a table, processes the data block to convert to the values in the objects instance variables by name based on the list in the "DATA" block which has been run through "TTF_Init_Fields"

`TTF_Unpack($fmt, $dat)`

A TrueType types equivalent of Perl's "unpack" function. Thus \$fmt consists of type followed by an optional number of elements to read including *. The type may be one of:

- c BYTE
- C CHAR
- f FIXED
- F F2DOT14
- I LONG
- L ULONG
- s SHORT
- S USHORT
- v Version number (FIXED)

Note that "FUNIT", "FWORD" and "UFWORD" are not data types but units.

Returns array of scalar (first element) depending on context

`$dat = TTF_Out_Fields($obj, $fields, $len)`

Given the fields table from "TTF_Init_Fields" writes out the instance variables from the object to the filehandle in TTF binary form.

`$dat = TTF_Pack($fmt, @data)`

The TrueType equivalent to Perl's "pack" function. See details of "TTF_Unpack" for how to work the \$fmt string.

`($num, $range, $select, $shift) = TTF_bininfo($num)`

Calculates binary search information from a number of elements

`TTF_word_utf8($str)`

Returns the UTF8 form of the 16 bit string, assumed to be in big endian order, including surrogate handling

TTF_utf8_word(\$str)

Returns the 16-bit form in big endian order of the UTF 8 string, including surrogate handling to Unicode.

XML_hexdump(\$context, \$dat)

Dumps out the given data as a sequence of <data> blocks each 16 bytes wide

XML_outhints

Converts a binary string of hinting code into a textual representation

make_circle(\$f, \$cmap, [\$dia, \$sb, \$opts])

Adds a dotted circle to a font. This function is very configurable. The parameters passed in are:

\$f Font to work with. This is required.

\$cmap

A cmap table (not the 'val' sub-element of a cmap) to add the glyph too.

Optional.

\$dia

Optional diameter for the main circle. Defaults to 80% em

\$sb Side bearing. The left and right side-bearings are always the same. This value defaults to 10% em.

There are various options to control all sorts of interesting aspects of the circle

numDots

Number of dots in the circle

numPoints

Number of curve points to use to create each dot

uid Unicode reference to store this glyph under in the cmap. Defaults to 0x25CC

pname

Postscript name to give the glyph. Defaults to uni25CC.

-dRadius

Radius of each dot.

BUGS

No known bugs

AUTHOR

Martin Hosken <<http://scripts.sil.org/FontUtils>>.

LICENSING

Copyright (c) 1998-2016, SIL International (<http://www.sil.org>)

This module is released under the terms of the Artistic License 2.0. For details, see the full text of the license in the file LICENSE.

perl v5.24.1

2016-08-28

Font::TTF::Utils(3pm)