

NAME

Font::TTF::Table – Superclass for tables and used for tables we don't have a class for

DESCRIPTION

Looks after the purely table aspects of a TTF table, such as whether the table has been read before, locating the file pointer, etc. Also copies tables from input to output.

INSTANCE VARIABLES

Instance variables start with a space

`read`

Flag which indicates that the table has already been read from file.

`dat` Allows the creation of unspecific tables. Data is simply output to any font file being created.

`nocompress`

If set, overrides the font default for WOFF table compression. Is a scalar integer specifying a table size threshold below which this table will not be compressed. Set to `-1` to never compress; `0` to always compress.

`INFILE`

The read file handle

`OFFSET`

Location of the file in the input file

`LENGTH`

Length in the input directory

`ZLENGTH`

Compressed length of the table if a WOFF font. $0 < ZLENGTH < LENGTH$ implies table is compressed.

`CSUM`

Checksum read from the input file's directory

`PARENT`

The Font::TTF::Font that table is part of

METHODS**Font::TTF::Table->new(%parms)**

Creates a new table or subclass. Table instance variables are passed in at this point as an associative array.

\$t->read

Reads the table from the input file. Acts as a superclass to all true tables. This method marks the table as read and then just sets the input file pointer but does not read any data. If the table has already been read, then returns `undef` else returns `$self`

For WOFF-compressed tables, the table is first decompressed and a replacement file handle is created for reading the decompressed data. In this case `ORIGINALOFFSET` will preserve the original value of `OFFSET` for applications that care.

\$t->read_dat

Reads the table into the `dat` instance variable for those tables which don't know any better

\$t->out(\$fh)

Writes out the table to the font file. If there is anything in the `dat` instance variable then this is output, otherwise the data is copied from the input file to the output

\$t->out_xml(\$context)

Outputs this table in XML format. The table is first read (if not already read) and then if there is no subclass, then the data is dumped as hex data

\$t->XML_element

Output a particular element based on its contents.

`$t->XML_end($context, $tag, %attrs)`

Handles the default type of `<data>` for those tables which aren't subclassed

`$t->minsize()`

Returns the minimum size this table can be. If it is smaller than this, then the table must be bad and should be deleted or whatever.

`$t->dirty($val)`

This sets the dirty flag to the given value or 1 if no given value. It returns the value of the flag

`$t->update`

Each table knows how to update itself. This consists of doing whatever work is required to ensure that the memory version of the table is consistent and that other parameters in other tables have been updated accordingly. I.e. by the end of sending `update` to all the tables, the memory version of the font should be entirely consistent.

Some tables which do no work indicate to themselves the need to update themselves by setting `isDirty` above 1. This method resets that accordingly.

`$t->empty`

Clears a table of all data to the level of not having been read

`$t->release`

Releases ALL of the memory used by this table, and all of its component/child objects. This method is called automatically by `Font::TTF::Font->release` (so you don't have to call it yourself).

NOTE, that it is important that this method get called at some point prior to the actual destruction of the object. Internally, we track things in a structure that can result in circular references, and without calling `release()` these will not properly get cleaned up by Perl. Once this method has been called, though, don't expect to be able to do anything with the `Font::TTF::Table` object; it'll have **no** internal state whatsoever.

Developer note: As part of the brute-force cleanup done here, this method will throw a warning message whenever unexpected key values are found within the `Font::TTF::Table` object. This is done to help ensure that any unexpected and unfreed values are brought to your attention so that you can bug us to keep the module updated properly; otherwise the potential for memory leaks due to dangling circular references will exist.

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`.