NAME

Font::TTF::Kern – Kerning tables

DESCRIPTION

Kerning tables are held as an ordered collection of subtables each giving incremental information regarding the kerning of various pairs of glyphs.

The basic structure of the kerning data structure is:

```
$kern = $f->{'kern'}{'tables'}[$tnum]{'kerns'}{$leftnum}{$rightnum};
```

Due to the possible complexity of some kerning tables the above information is insufficient. Reference also needs to be made to the type of the table and the coverage field.

INSTANCE VARIABLES

The instance variables for a kerning table are relatively straightforward.

Version
: Version number of the kerning table

Num
: Number of subtables in the kerning table

tables
: Array of subtables in the kerning table

  Each subtable has a number of instance variables.

  kern
  : A two level hash array containing kerning values. The indexing is left is via left class and right class. It may seem using hashes is strange, but most tables are not type 2 and this method saves empty array values.

  type
  : Stores the table type. Only type 0 and type 2 tables are specified for TrueType so far.

  coverage
  : A bit field of coverage information regarding the kerning value. See the TrueType specification for details.

  Version
  : Contains the version number of the table.

  Num
  : Number of kerning pairs in this type 0 table.

  left An array indexed by glyph – left_first which returns a class number for the glyph in type 2 tables.

  right
  : An array indexed by glyph – right_first which returns a class number for the glyph in type 2 tables.

  left_first
  : the glyph number of the first element in the left array for type 2 tables.

  right_first
  : the glyph number of the first element in the right array for type 2 tables.

  num_left
  : Number of left classes

  num_right
  : Number of right classes

METHODS

`$t`**->read**
Reads the whole kerning table into structures

`$t`**->out($fh)**
Outputs the kerning tables to the given file

`$t`**->XML_element($context,** `$depth`**,** `$key`**,** `$value`**)**
Handles outputting the kern hash into XML a little more tidily

`$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.

BUGS

- Only supports kerning table types 0 & 2.
- No real support functions to *do* anything with the kerning tables yet.

AUTHOR

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

LICENSING