NAME

Font::TTF::GPOS − Support for Opentype GPOS tables in conjunction with TTOpen

DESCRIPTION

The GPOS table is one of the most complicated tables in the TTF spec and the corresponding data structure abstraction is also not trivial. While much of the structure of a GPOS is shared with a GSUB table via the Font::TTF::Ttopen

INSTANCE VARIABLES

Here we describe the additions and lookup specific information for GPOS tables. Unfortunately there is no one abstraction which seems to work comfortable for all GPOS tables, so we will also examine how the variables are used for different lookup types.

The following are the values allowed in the ACTION_TYPE and MATCH_TYPE variables:

ACTION_TYPE

This can take any of the following values

a The ACTION is an array of anchor tables

o Offset. There is no RULE array. The ADJUST variable contains a value record (see later in this description)

v The ACTION is a value record.

p Pair adjustment. The ACTION contains an array of two value records for the matched two glyphs.

e Exit and Entry records. The ACTION contains an array of two anchors corresponding to the exit and entry anchors for the glyph.

l Indicates a lookup based contextual rule as per the GSUB table.

MATCH_TYPE

This can take any of the following values

g A glyph array

c An array of class values

o An array of coverage tables

The following variables are added for Attachment Positioning Subtables:

MATCH

This contains an array of glyphs to match against for all RULES. It is much like having the same MATCH string in all RULES. In the cases it is used so far, it only ever contains one element.

MARKS

This contains a Mark array consisting of each element being a subarray of two elements:

CLASS The class that this mark uses on its base

ANCHOR

The anchor with which to attach this mark glyph

The base table for mark to base, ligature or mark attachment positioning is structured with the ACTION containing an array of anchors corresponding to each attachment class. For ligatures, there is more than one RULE in the RULE array corresponding to each glyph in the coverage table.

Other variables which are provided for informational purposes are:

VFMT

Value format for the adjustment of the glyph matched by the coverage table.

VFMT2

Value format used in pair adjustment for the second glyph in the pair

Value Records

There is a subtype used in GPOS tables called a value record. It is used to adjust the position of a glyph from its default position. The value record is variable length with a bitfield at the beginning to indicate which of the following entries are included. The bitfield is not stored since it is recalculated at write time.

XPlacement
: Horizontal adjustment for placement (not affecting other unattached glyphs)

YPlacement
: Vertical adjustment for placement (not affecting other unattached glyphs)

XAdvance
: Adjust the advance width glyph (used only in horizontal writing systems)

YAdvance
: Adjust the vertical advance (used only in vertical writing systems)

XPlaDevice
: Device table for device specific adjustment of horizontal placement

YPlaDevice
: Device table for device specific adjustment of vertical placement

XAdvDevice
: Device table for device specific adjustment of horizontal advance

YAdDevice
: Device table for device specific adjustment of vertical advance

XIdPlacement
: Horizontal placement metric id (for Multiple Master fonts – but that is all I know!)

YIdPlacement
: Vertical placement metric id

XIdAdvance
: Horizontal advance metric id

YIdAdvance
: Vertical advance metric id

CORRESPONDANCE TO LAYOUT TYPES

Here is what is stored in the ACTION_TYPE and MATCH_TYPE for each of the known GPOS subtable types:

	1.1	1.2	2.1	2.2	3	4	5	6	7.1	7.2	7.3	8.1	8.2	8.3
ACTION_TYPE	o	v	p	p	e	a	a	a	l	l	l	l	l	l
MATCH_TYPE			g	c					g	c	o	g	c	o

METHODS

read_sub

Reads the subtable into the data structures

`$t`**->extension**

Returns the table type number for the extension table

`$t`**->out_sub**

Outputs the subtable to the given filehandle

`$t`**->read_value($format,** `$base`**,** `$lookup`**,** `$fh`**)**

Reads a value record from the current location in the file, according to the format given.

`$t`**->read_delta($offset,** `$base`**,** `$lookup`**,** `$fh`**)**

Reads a delta (device table) at the given offset if it hasn not already been read. Store the offset and item in the lookup cache ($lookup->{' CACHE'})

`$t`**->read_anchor($offset,** `$base`**,** `$lookup`**,** `$fh`**)**
Reads an Anchor table at the given offset if it has not already been read.

`$t`**->fmt_value**
Returns the value format for a given value record

`$t`**->out_value**
Returns the output string for the outputting of the value for a given format. Also updates the offset cache for any device tables referenced.

AUTHOR

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

LICENSING