

NAME

Font::TTF::Coverage – OpenType coverage and class definition objects

DESCRIPTION

Coverage tables and class definition objects are virtually identical concepts in OpenType. Their difference comes purely in their storage. Therefore we can say that a coverage table is a class definition in which the class definition for each glyph is the corresponding index in the coverage table. The resulting data structure is that a Coverage table has the following fields:

cover

A boolean to indicate whether this table is a coverage table (TRUE) or a class definition (FALSE)

val A hash of glyph ids against values (either coverage index or class value)

fmt The storage format used is given here, but is recalculated when the table is written out.

count

A count of the elements in a coverage table for use with `add`. Each subsequent addition is added with the current count and increments the count.

max

Maximum class value in a class table.

METHODS**new(\$isCover [, vals])**

Creates a new coverage table or class definition table, depending upon the value of `$isCover`. if `$isCover` then `vals` may be a list of glyphs to include in order. If no `$isCover`, then `vals` is a hash of glyphs against class values.

read(\$fh)

Reads the coverage/class table from the given file handle

out(\$fh, \$state)

Writes the coverage/class table to the given file handle. If `$state` is 1 then the output string is returned rather than being output to a filehandle.

\$c->add(\$glyphid[, \$class])

Adds a glyph id to the coverage or class table. Returns the index or class number of the glyphid added.

\$c->signature

Returns a vector of all the glyph ids covered by this coverage table or class

@map=\$c->sort

Sorts the coverage table so that indexes are in ascending order of glyphid. Returns a map such that `$map[$new_index]=$old_index`.

\$c->out_xml(\$context)

Outputs this coverage/class in XML

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.