NAME

Font::TTF::Cmap – Character map table

DESCRIPTION

Looks after the character map. For ease of use, the actual cmap is held in a hash against codepoint. Thus for a given table:

```
$gid = $font->{'cmap'}{'Tables'}[0]{'val'}{$code};
```

Note that `$code` should be a true value (0x1234) rather than a string representation.

INSTANCE VARIABLES

The instance variables listed here are not preceded by a space due to their emulating structural information in the font.

Num
: Number of subtables in this table

Tables
: An array of subtables ([0..Num−1])

  Each subtable also has its own instance variables which are, again, not preceded by a space.

  Platform
  : The platform number for this subtable

  Encoding
  : The encoding number for this subtable

  Format
  : Gives the stored format of this subtable

  Ver
  : Gives the version (or language) information for this subtable

  val
  : A hash keyed by the codepoint value (not a string) storing the glyph id

The following cmap options are controlled by instance variables that start with a space:

allowholes
: By default, when generating format 4 cmap subtables character codes that point to glyph zero (normally called .notdef) are not included in the subtable. In some cases including some of these character codes can result in a smaller format 4 subtable. To enable this behavior, set allowholes to non-zero.

METHODS

`$t`**->read**
: Reads the cmap into memory. Format 4 subtables read the whole subtable and fill in the segmented array accordingly.

`$t`**->ms_lookup($uni)**
: Finds a Unicode table, giving preference to the MS one, and looks up the given Unicode codepoint in it to find the glyph id.

`$t`**->find_ms**
: Finds the a Unicode table, giving preference to the Microsoft one, and sets the `mstable` instance variable to it if found. Returns the table it finds.

`$t`**->ms_enc**
: Returns the encoding of the microsoft table (0 => symbol, etc.). Returns undef if there is no Microsoft cmap.

`$t`**->out($fh)**
: Writes out a cmap table to a filehandle. If it has not been read, then just copies from input file to output

`$t`**->XML_element($context,** `$depth`**,** `$name`**,** `$val`**)**
Outputs the elements of the cmap in XML. We only need to process val here

`$t`**->***minsize()*
Returns the minimum size this table can be in bytes. If it is smaller than this, then the table must be bad and should be deleted or whatever.

`$t`**->update**
Tidies the cmap table.

Removes MS Fmt12 cmap if it is no longer needed.

Removes from all cmaps any codepoint that map to GID=0. Note that such entries will be re-introduced as necessary depending on the cmap format.

`@map` **=** `$t`**->reverse(%opt)**
Returns a reverse map of the Unicode cmap. I.e. given a glyph gives the Unicode value for it. Options are:

tnum
Table number to use rather than the default Unicode table

array
Returns each element of reverse as an array since a glyph may be mapped by more than one Unicode value. The arrays are unsorted. Otherwise store any one unicode value for a glyph.

is_unicode($index)
Returns whether the table of a given index is known to be a unicode table (as specified in the specifications)

BUGS

- Format 14 (Unicode Variation Sequences) cmaps are not supported.

AUTHOR

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

LICENSING