NAME

Font::TTF::Name − String table for a TTF font

DESCRIPTION

Strings are held by number, platform, encoding and language. Strings are accessed as:

```
    $f->{'name'}{'strings'}[$number][$platform_id][$encoding_id]{$language_id}
```

Notice that the language is held in an associative array due to its sparse nature on some platforms such as Microsoft ($pid = 3). Notice also that the array order is different from the stored array order (platform, encoding, language, number) to allow for easy manipulation of strings by number (which is what I guess most people will want to do).

By default, `$Font::TTF::Name::utf8` is set to 1, and strings will be stored as UTF8 wherever possible. The method `is_utf8` can be used to find out if a string in a particular platform and encoding will be returned as UTF8. Unicode strings are always converted if utf8 is requested. Otherwise, strings are stored according to platform:

You now have to set <$Font::TTF::Name::utf8> to 0 to get the old behaviour.

Apple Unicode (platform id = 0)
: Data is stored as network ordered UCS2. There is no encoding id for this platform but there are language ids as per Mac language ids.

Mac (platform id = 1)
: Data is stored as 8−bit binary data, leaving the interpretation to the user according to encoding id.

Unicode (platform id = 2)
: Currently stored as 16−bit network ordered UCS2. Upon release of Perl 5.005 this will change to utf8 assuming current UCS2 semantics for all encoding ids.

Windows (platform id = 3)
: As per Unicode, the data is currently stored as 16−bit network ordered UCS2. Upon release of Perl 5.005 this will change to utf8 assuming current UCS2 semantics for all encoding ids.

INSTANCE VARIABLES

strings
: An array of arrays, etc.

METHODS

`$t`−>**read**
: Reads all the names into memory

`$t`−>**out($fh)**
: Writes out all the strings

`$t`−>**XML_element($context,** `$depth`**,** `$key`**,** `$value`**)**
: Outputs the string element in nice XML (which is all the table really!)

`$t`−>**XML_end($context,** `$tag`**,** `%attrs`**)**
: Store strings in the right place

`$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.

is_utf8($pid, `$eid`**)**
: Returns whether a string of a given platform and encoding is going to be in UTF8

find_name($nid)
: Hunts down a name in all the standard places and returns the string and for an array context the pid, eid & lid as well

remove_name($nid)

Removes all strings with the given name id from the table.

set_name($nid, `$str`**[,** `$lang`**[,** `@cover`**]])**

Sets the given name id string to `$str` for all platforms and encodings that this module can handle. If `$lang` is set, it is interpretted as a language tag and if the particular language of a string is found to match, then that string is changed, otherwise no change occurs.

If supplied, `@cover` should be a list of references to two-element arrays containing pid,eid pairs that should be added to the name table if not already present.

This function does not add any names to the table unless `@cover` is supplied.

Font::TTF::Name−>match_lang($pid, `$lid`**,** `$lang`**)**

Compares the language associated to the string of given platform and language with the given language tag. If the language matches the tag (i.e. is equal or more defined than the given language tag) returns true. This is calculated by finding whether the associated language tag starts with the given language tag.

Font::TTF::Name−>get_lang($pid, `$lid`**)**

Returns the language tag associated with a particular platform and language id

Font::TTF::Name−>find_lang($pid, `$lang`**)**

Looks up the language name and returns a lang id if one exists

Font::TTF::Name−>*pe_list()*

Returns an array of references to two-element arrays containing pid,eid pairs that already exist in this name table. Useful for creating `@cover` parameter to *set_name()*.

BUGS

- Unicode type strings will be stored in utf8 for all known platforms, once Perl 5.6 has been released and I can find all the mapping tables, etc.

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.