POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

Charset.pm

Table of Contents
Charset.pm

NAME

MARC::Charset - A module for doing MARC-8/UTF8 translation

SYNOPSIS

```
use MARC::Charset;

## create a MARC::Charset object
my $charset = MARC::Charset->new();

## a string containing the Ansel value for a copyright symbol
my $ansel = chr(0xC3) . ' copyright 1969'.

## the same string, but now encoded in UTF8!
my $utf8 = $charset->to_utf8($extLatin);
```

DESCRIPTION

MARC::Charset is a package that allows you to easily convert between the MARC-8 character encodings and Unicode (UTF-8). The Library of Congress maintains some essential mapping tables and information about the MARC-8 and Unicode environments at:

```
http://www.loc.gov/marc/specifications/spechome.html
```

MARC::Charset is essentially a Perl implementation of the specifications found at LC, and supports the following character sets:

- Latin (Basic/Extended + Greek Symbols, Subscripts and Superscripts)
- Hebrew
- Cyrillic (Basic + Extended)
- Arabic (Basic + Extended)
- Greek

Since the East Asian character set is 32 bit, there isn't support just yet in MARC::Charset for them. It's been built with an eye for the future, and so when more is understood about how 32 bit graphical character sets are designated as working G0 and G1 sets, then more will be done.

METHODS

new()

The constructor which will return MARC::Charset object. If you like you can pass in the default G0 and G1 charsets (using the g0 and g1 parameters, but if you don't ASCII/Ansel will be assumed.

```
## for standard characters sets: ASCII and Ansel
my $cs = MARC::Charset->new();

## or if you want to specify Arabic Basic + Extended as the G0/G1 character
## sets.
my $cs = MARC::Charset->new(
   g0 => MARC::Charset::ArabicBasic->new(),
   g1 => MARC::Charset::ArabicExtended->new()
);
```

If you would like diagnostics turned on pass in the DIAGNOSTICS parameter and set it to a value that will evaluate to true (eg. 1).

```
my $cs = MARC::Charset->new( diagnostics => 1 );
```

to_utf8()

Pass to_utf8() a string of MARC8 encoded characters and get back a string of UTF8 characters. to_utf8() will handle escape sequences within the string that change the working character sets to Greek, Hebrew, Arabic (Basic + Extended), Cyrillic (Basic + Extended)...but not 32 bit East Asian (see TODO).

g0()

Returns an object representing the character set that is being used as the first graphic character set (G0). If you pass in a MARC::Charset::* object you will set the G0 character set, and as a side effect you'll

get the previous G0 value returned to you. You probably don't ever need to call this since character set changes are handled when you call to_utf8(), but it's here if you want it.

```
## set the G0 character set to Greek
my $charset = MARC::Charset->new();
$charset->g0( MARC::Charset::Greek->new() );
```

g1()

Same as g0() above, but operates on the second graphic set that is available.

TODO

- to_marc8()

  A function for going from Unicode to MARC-8 character encodings.

- Support for 32bit MARC-8 characters:

  This concerns the East Asian character sets: Han, Hiragana, Katakana, Hangul and Punctuation. I'm a bit confused about whether 7/8 bit character sets can interoperate with 32 bit character sets. For example if ASCII is designated as the working G0 character set, and East Asian as the working G1 character set. While I've tried to program towards supporting 32 bit character sets I need to know exactly how they are implemented in the 'real world'. So if you have any East Asian MARC data please email it to me!!

SEE ALSO

MARC::Charset::ASCII
MARC::Charset::Ansel
MARC::Charset::ArabicBasic
MARC::Charset::ArabicExtended
MARC::Charset::Controls
MARC::Charset::CyrillicBasic
MARC::Charset::CyrillicExtended
MARC::Charset::Greek
MARC::Charset::GreekSymbols
MARC::Charset::Hebrew
MARC::Charset::Subscripts
MARC::Charset::Superscripts

VERSION HISTORY

- v.01 - 2002.07.17 (ehs)

AUTHORS

Ed Summers <ehs@pobox.com>