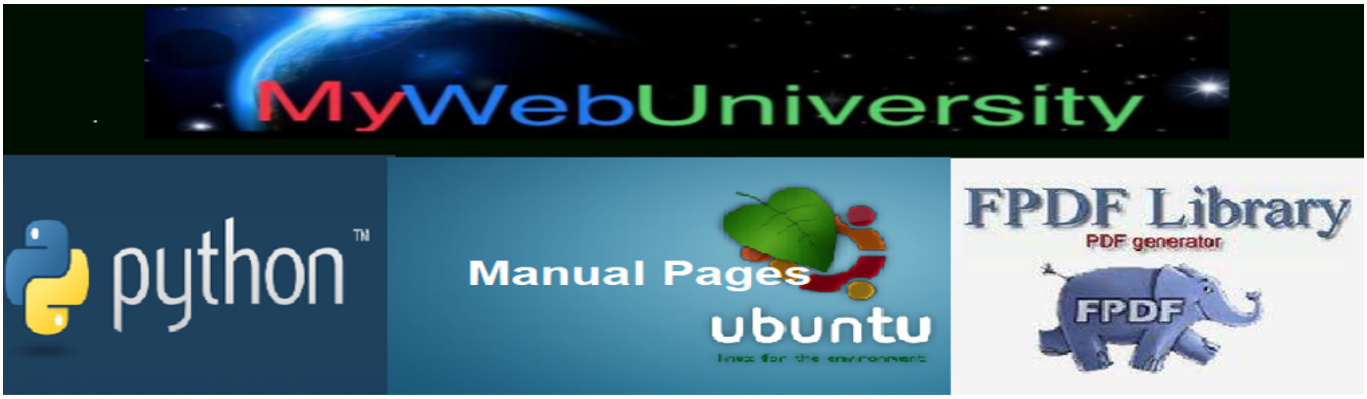

Rocky Enterprise Linux 9.2 Manual Pages on command 'Encode::GSM0338.3perl'

$ man Encode::GSM0338.3perl

NAME

Encode::GSM0338 -- ETSI GSM 03.38 Encoding

SYNOPSIS

```
use Encode qw/encode decode/;
$gsm0338 = encode("gsm0338", $unicode); # loads Encode::GSM0338 implicitly
$unicode = decode("gsm0338", $gsm0338); # ditto
```

DESCRIPTION

GSM0338 is for GSM handsets. Though it shares alphanumerals with ASCII, control character ranges and other parts are mapped very differently, mainly to store Greek characters.

There are also escape sequences (starting with 0x1B) to cover e.g. the Euro sign.

This was once handled by Encode::Bytes but because of all those unusual specifications, Encode 2.20 has relocated the support to this module.

This module implements only GSM 7 bit Default Alphabet and GSM 7 bit default alphabet extension table according to standard 3GPP TS 23.038 version 16. Therefore National Language Single Shift and National Language Locking Shift are not implemented nor supported.

Septets

This modules operates with octets (like any other Encode module) and not with packed septets (unlike other GSM standards). Therefore for processing binary SMS or parts of GSM TPDU payload (3GPP TS 23.040) it is needed to do conversion between octets and packed septets. For this purpose perl's "pack" and "unpack" functions may be useful:

```
$bytes = substr(pack('(b*)*', unpack '(A7)*', unpack 'b*', $septets), 0, $num_of_septets);
$unicode = decode('GSM0338', $bytes);

$bytes = encode('GSM0338', $unicode);
$septets = pack 'b*', join '', map { substr $_, 0, 7 } unpack '(A8)*', unpack 'b*', $bytes;
$num_of_septets = length $bytes;
```

Please note that for correct decoding of packed septets it is required to know number of septets packed in binary buffer as binary buffer is always padded with zero bits and 7 zero bits represents character "@". Number of septets is also stored in TPDU payload when dealing with 3GPP TS 23.040.

BUGS

Encode::GSM0338 2.7 and older versions (part of Encode 3.06) incorrectly handled zero bytes (character "@"). This was fixed in Encode::GSM0338 version 2.8 (part of Encode 3.07).

SEE ALSO

3GPP TS 23.038 <https://www.3gpp.org/dynareport/23038.htm>

ETSI TS 123 038 V16.0.0 (2020-07) <https://www.etsi.org/deliver/etsi_ts/123000_123099/123038/16.00.00_60/ts_123038v160000p.pdf>

Encode