

## NAME

Encode::GSM0338 -- ESTI GSM 03.38 Encoding

## SYNOPSIS

```
use Encode qw/encode decode/;
$gsm0338 = encode("gsm0338", $utf8);    # loads Encode::GSM0338
implicitly
$utf8    = 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.

## NOTES

Unlike most other encodings, the following always croaks on error for any *\$chk* that evaluates to true.

```
$gsm0338 = encode("gsm0338", $utf8, $chk);
$utf8    = decode("gsm0338", $gsm0338, $chk);
```

So if you want to check the validity of the encoding, surround the expression with `eval { }` block as follows;

```
eval {
    $utf8    = decode("gsm0338", $gsm0338, $chk);
};
if ($@){
    # handle exception here
}
```

## BUGS

ESTI GSM 03.38 Encoding itself.

Mapping `\x00` to `'@'` causes too much pain everywhere.

Its use of `\x1b` (escape) is also very questionable.

Because of those two, the code paging approach used use in ucm-based Encoding *SOMETIMES* fails so this module was written.

## SEE ALSO

*Encode*