NAME

Text::Soundex - Implementation of the soundex algorithm.

SYNOPSIS

```
  use Text::Soundex;

  # Original algorithm.
  $code = soundex($name);    # Get the soundex code for a name.
  @codes = soundex(@names);  # Get the list of codes for a list of names.

  # American Soundex variant (NARA) - Used for US census data.
  $code = soundex_nara($name);    # Get the soundex code for a name.
  @codes = soundex_nara(@names);  # Get the list of codes for a list of
names.

  # Redefine the value that soundex() will return if the input string
  # contains no identifiable sounds within it.
  $Text::Soundex::nocode = 'Z000';
```

DESCRIPTION

Soundex is a phonetic algorithm for indexing names by sound, as pronounced in English. The goal is for names with the same pronunciation to be encoded to the same representation so that they can be matched despite minor differences in spelling. Soundex is the most widely known of all phonetic algorithms and is often used (incorrectly) as a synonym for "phonetic algorithm". Improvements to Soundex are the basis for many modern phonetic algorithms. (Wikipedia, 2007)

This module implements the original soundex algorithm developed by Robert Russell and Margaret Odell, patented in 1918 and 1922, as well as a variation called "American Soundex" used for US census data, and current maintained by the National Archives and Records Administration (NARA).

The soundex algorithm may be recognized from Donald Knuth's **The Art of Computer Programming** . The algorithm described by Knuth is the NARA algorithm.

The value returned for strings which have no soundex encoding is defined using `$Text::Soundex::nocode`. The default value is `undef`, however values such as `'Z000'` are commonly used alternatives.

For backward compatibility with older versions of this module the `$Text::Soundex::nocode` is exported into the caller's namespace as `$soundex_nocode`.

In scalar context, `soundex()` returns the soundex code of its first argument. In list context, a list is returned in which each element is the soundex code for the corresponding argument passed to `soundex()`. For example, the following code assigns @codes the value `('M200', 'S320')`:

```
   @codes = soundex qw(Mike Stok);
```

To use `Text::Soundex` to generate codes that can be used to search one of the publically available US Censuses, a variant of the soundex algorithm must be used:

```
    use Text::Soundex;
    $code = soundex_nara($name);
```

An example of where these algorithm differ follows:

```
    use Text::Soundex;
    print soundex("Ashcraft"), "\n";       # prints: A226
```

```
print soundex_nara("Ashcraft"), "\n";  # prints: A261
```

EXAMPLES

Donald Knuth's examples of names and the soundex codes they map to are listed below:

```
Euler, Ellery -> E460
Gauss, Ghosh -> G200
Hilbert, Heilbronn -> H416
Knuth, Kant -> K530
Lloyd, Ladd -> L300
Lukasiewicz, Lissajous -> L222
```

so:

```
$code = soundex 'Knuth';         # $code contains 'K530'
@list = soundex qw(Lloyd Gauss); # @list contains 'L300', 'G200'
```

LIMITATIONS

As the soundex algorithm was originally used a **long** time ago in the US it considers only the English alphabet and pronunciation. In particular, non-ASCII characters will be ignored. The recommended method of dealing with characters that have accents, or other unicode characters, is to use the Text::Unidecode module available from CPAN. Either use the module explicitly:

```
use Text::Soundex;
use Text::Unidecode;

print soundex(unidecode("Fran\xE7ais")), "\n"; # Prints "F652\n"
```

Or use the convenient wrapper routine:

```
use Text::Soundex 'soundex_unicode';

print soundex_unicode("Fran\xE7ais"), "\n";    # Prints "F652\n"
```

Since the soundex algorithm maps a large space (strings of arbitrary length) onto a small space (single letter plus 3 digits) no inference can be made about the similarity of two strings which end up with the same soundex code. For example, both `Hilbert` and `Heilbronn` end up with a soundex code of `H416`.

MAINTAINER

This module is currently maintain by Mark Mielke (`mark@mielke.cc`).

HISTORY

Version 3 is a significant update to provide support for versions of Perl later than Perl 5.004. Specifically, the XS version of the soundex() subroutine understands strings that are encoded using UTF-8 (unicode strings).

Version 2 of this module was a re-write by Mark Mielke (`mark@mielke.cc`) to improve the speed of the subroutines. The XS version of the soundex() subroutine was introduced in 2.00.

Version 1 of this module was written by Mike Stok (`mike@stok.co.uk`) and was included into the Perl core library set.

Dave Carlsen (`dcarlsen@csranet.com`) made the request for the NARA algorithm to be included. The NARA soundex page can be viewed at:

http://www.nara.gov/genealogy/soundex/soundex.htmlIan Phillips (ian@pipex.net) and Rich Pinder (rpinder@hsc.usc.edu) supplied ideas and spotted mistakes for v1.x.