

metaphone egg

Metaphone 'sounds like' algorithm
Extension for Chicken Scheme
Version 0.1

Kon Lovett

Table of Contents

1	About this egg	1
1.1	Version history	1
1.2	Requirements	1
1.3	Usage	1
2	Documentation	2
3	License	4
	Index	5

1 About this egg

1.1 Version history

0.1 Initial release

1.2 Requirements

This egg requires the following extensions:

`miscmacros`, `misc-extn`

1.3 Usage

Load this egg like so:

- `(require-extension srfi-13 metaphone)`
- or
- `(require-extension utf8 utf8-srfi-13 metaphone)`
- `(import utf8)`
- `(import utf8-srfi-13)`

2 Documentation

Supplies an implementation of Lawrence Philips' Double Metaphone algorithm. A 'sounds like' string -> search key utility, similar to Soundex but with better match and worse computational performance. Strings must conform to English spelling yet are not limited to English words.

make-metaphone-key [procedure]

(make-metaphone-key TARGET [#:keylen 4] [#:binary? #t])

Returns an object comprising the Double Metaphone interpretation of the **TARGET** string.

keylen: A fixnum. The length of the result key.

binary?: A boolean. When the key length is 4 an optimization is possible, resulting in a fixnum key representation and corresponding comparison and space savings.

Binary keys can only be compared with other binary keys. Do not use the binary key option unless the programmer is prepared to ensure this situation.

metaphone-key=? [procedure]

(metaphone-key=? KEY1 KEY2)

Are the keys equivalent on the primary or alternate.

metaphone-key<>? [procedure]

(metaphone-key<>? KEY1 KEY2)

Guess.

metaphone-key-binary? [procedure]

(metaphone-key-binary? KEY)

Was binary encoding used to construct this metaphone key?

metaphone-key? [procedure]

(metaphone-key? OBJECT)

Guess?

metaphone-key-binary? [procedure]

(metaphone-key-binary? METAPHONE-KEY)

Is the key binary encoded? Only possible when key length is 4.

metaphone-key-source [procedure]

(metaphone-key-source METAPHONE-KEY)

Original string.

metaphone-key-length [procedure]

(metaphone-key-length METAPHONE-KEY)

Key length. When not binary the length of string key(s).

`metaphone-key-primary` [procedure]

(`metaphone-key-primary` METAPHONE-KEY)

The primary key, a string.

Note that when binary keys the result is a fixnum, not a string.

`metaphone-key-alternate` [procedure]

(`metaphone-key-alternate` METAPHONE-KEY)

The alternate key, a string.

Note that when binary keys the result is a fixnum, not a string.

`metaphone-key-with-alternate?` [procedure]

(`metaphone-key-with-alternate?` METAPHONE-KEY)

Is there an alternate key?

3 License

Copyright (c) 2006, Kon Lovett. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ASIS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Index

make-metaphone-key.....	2	metaphone-key-source.....	2
metaphone-key-alternate.....	3	metaphone-key-with-alternate?.....	3
metaphone-key-binary?.....	2	metaphone-key<>?.....	2
metaphone-key-length.....	2	metaphone-key=?.....	2
metaphone-key-primary.....	3	metaphone-key?.....	2