



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction
[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools
[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Cite this page](#)

Print/export
[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Languages
[Add links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Metaphone

From Wikipedia, the free encyclopedia

Lawrence Philips redirects here. For the football player, see *Lawrence Phillips*.

Metaphone is a [phonetic algorithm](#), published by Lawrence Philips in 1990, for indexing words by their English pronunciation.^[1] It fundamentally improves on the [Soundex](#) algorithm by using information about variations and inconsistencies in English spelling and pronunciation to produce a more accurate encoding, which does a better job of matching words and names which sound similar. As with Soundex, similar sounding words should share the same keys. Metaphone is available as a built-in operator in a number of systems.

The original author later produced a new version of the algorithm, which he named [Double Metaphone](#). Contrary to the original algorithm whose application is limited to English only, this version takes into account spelling peculiarities of a number of other languages. In 2009 Lawrence Philips released a third version, called Metaphone 3, which achieves an accuracy of approximately 99% for English words, non-English words familiar to Americans, and first names and family names commonly found in the United States, having been developed according to modern engineering standards against a test harness of prepared correct encodings.

Contents [hide]


- 1 Procedure
- 2 Double Metaphone
- 3 Metaphone 3
- 4 Common misconceptions
- 5 See also
- 6 External links
 - 6.1 Metaphone Algorithms For Other Languages
- 7 References

Procedure [edit]

Original Metaphone codes use the 16 [consonant](#) symbols 0BFHJKLMNPRSTWXY.^[2] The '0' represents *"th"* (as an [ASCII](#) approximation of Θ), 'X' represents *"sh"* or *"ch"*, and the others represent their usual English pronunciations. The [vowels](#) AEIOU are also used, but only at the beginning of the code.^[3] This table summarizes most of the rules in the original implementation:

- Drop duplicate adjacent letters, except for C.
- If the word begins with 'KN', 'GN', 'PN', 'AE', 'WR', drop the first letter.
- Drop 'B' if after 'M' at the end of the word.
- 'C' transforms to 'X' if followed by 'IA' or 'H' (unless in latter case, it is part of '-SCH-', in which case it transforms to 'K'). 'C' transforms to 'S' if followed by 'I', 'E', or 'Y'. Otherwise, 'C' transforms to 'K'.
- 'D' transforms to 'J' if followed by 'GE', 'GY', or 'G'. Otherwise, 'D' transforms to 'T'.
- Drop 'G' if followed by 'H' and 'H' is not at the end or before a vowel. Drop 'G' if followed by 'N' or 'NED' and is at the end.
- 'G' transforms to 'J' if before 'I', 'E', or 'Y', and it is not in 'GG'. Otherwise, 'G' transforms to 'K'.
- Drop 'H' if after vowel and not before a vowel.
- 'CK' transforms to 'K'.
- 'PH' transforms to 'F'.
- 'Q' transforms to 'K'.
- 'S' transforms to 'X' if followed by 'H', 'IO', or 'IA'.
- 'T' transforms to 'X' if followed by 'IA' or 'IO'. 'TH' transforms to '0'. Drop 'T' if followed by 'CH'.
- 'V' transforms to 'F'.
- 'WH' transforms to 'W' if at the beginning. Drop 'W' if not followed by a vowel.
- 'X' transforms to 'S' if at the beginning. Otherwise, 'X' transforms to 'KS'.
- Drop 'Y' if not followed by a vowel.
- 'Z' transforms to 'S'.
- Drop all vowels unless it is the beginning.

It should be noted, however, that this table does not constitute a complete description of the original Metaphone algorithm, and the algorithm cannot be coded correctly from it. Original Metaphone contained many errors and was superseded by Double Metaphone, and in turn Double Metaphone and original Metaphone were superseded by Metaphone 3, which corrects thousands of miscodings that will be produced by the first two versions.

To implement Metaphone without purchasing a (source code) copy of Metaphone 3, the best guide would be the reference implementation of Double Metaphone, which may be found [here](#) .

Double Metaphone [\[edit\]](#)

The **Double Metaphone** phonetic encoding algorithm is the second generation of this algorithm. Its implementation was described in the June 2000 issue of *C/C++ Users Journal*. It makes a number of fundamental design improvements over the original Metaphone algorithm.

It is called "Double" because it can return both a primary and a secondary code for a string; this accounts for some ambiguous cases as well as for multiple variants of surnames with common ancestry. For example, encoding the name "Smith" yields a primary code of *SMO* and a secondary code of *XMT*, while the name "Schmidt" yields a primary code of *XMT* and a secondary code of *SMT*—both have *XMT* in common.

Double Metaphone tries to account for myriad irregularities in [English](#) of [Slavic](#), [Germanic](#), [Celtic](#), [Greek](#), [French](#), [Italian](#), [Spanish](#), [Chinese](#), and other origin. Thus it uses a much more complex ruleset for coding than its predecessor; for example, it tests for approximately 100 different contexts of the use of the letter C alone.

Metaphone 3 [\[edit\]](#)

A professional version was released in October 2009, developed by the same author, Lawrence Philips. It is a commercial product but is sold as source code. Metaphone 3 further improves phonetic encoding of words in the English language, non-English words familiar to Americans, and first names and family names commonly found in the United States.^[4] It improves encoding for proper names in particular to a considerable extent.^[5] The author claims that in general it improves accuracy for all words from the approximately 89% of Double Metaphone to 98%. Developers can also now set switches in code to cause the algorithm to encode Metaphone keys 1) taking non-initial vowels into account, as well as 2) encoding voiced and unvoiced consonants differently. This allows the result set to be more closely focused if the developer finds that the search results include too many words that don't resemble the search term closely enough.^[6] Metaphone 3 is sold as C++, Java, C#, PHP, Perl, and PL/SQL source, Ruby and Python wrappers accessing a Java jar, and also Metaphone 3 for Spanish and German pronunciation available as Java and C# source.^[7] The latest revision of the Metaphone 3 algorithm is v2.5.4, released March 2015.

Common misconceptions [\[edit\]](#)

There are a couple of misconceptions about the Metaphone algorithms that should be addressed. The following statements are true:

1. All of them are designed to address regular, "dictionary" words, not just names, and
2. Metaphone algorithms do **not** produce phonetic representations of the input words and names; rather, the output is an intentionally **approximate** phonetic representation, according to this standard:
 - words that start with a vowel sound will have an 'A', representing any vowel, as the first character of the encoding (in Double Metaphone and Metaphone 3 - original Metaphone just preserves the actual vowel),
 - vowels after an initial vowel sound will be disregarded and not encoded, and
 - voiced/unvoiced consonant pairs will be mapped to the same encoding. (Examples of voiced/unvoiced consonant pairs are D/T, B/P, Z/S, G/K, etc.).

This approximate encoding is necessary to account for the way English speakers vary their pronunciations and misspell or otherwise vary words and names they are trying to spell. Vowels, of course, are notoriously highly variable. British speakers often complain that Americans seem to pronounce 'T's the same as 'D'. Consider, also, that all English speakers often pronounce 'Z' where 'S' is spelled, almost always when a noun ending in a voiced consonant or a liquid is pluralized, for example "seasons", "beams", "examples", etc. Not encoding vowels after an initial vowel sound will help to group words where a vowel and a consonant may be transposed in the misspelling or alternative pronunciation.

See also [\[edit\]](#)

- [Caverphone](#)
- [Soundex](#)
- [New York State Identification and Intelligence System](#)
- [Match Rating Approach](#)

External links [[edit](#)]

- [Another explanation of the algorithm](#)
- [Project to write a formal definition of the algorithm](#)
- ["The Double Metaphone Search Algorithm"](#), *C/C++ Users Journal*, June 2000 (full-text access requires registration)
- [The Double Metaphone Search Algorithm](#), By Lawrence Phillips, June 1, 2000, Dr Dobb's, *Original article*

Metaphone Algorithms For Other Languages [[edit](#)]

- [Brazilian Portuguese in C](#) Metaphone for Brazilian Portuguese, in C with PHP and PostgreSQL port.
- [Spanish Metaphone in Python](#)
- [Double Metaphone algorithm for Bangla](#)
- [Double Metaphone algorithm for Amharic](#)
- [Russian Metaphone in Ruby](#). Implemented by [CleverUA](#)
- [Metaphone 3 for Spanish and German](#)

References [[edit](#)]

- ↑ Hanging on the Metaphone, Lawrence Philips. Computer Language, Vol. 7, No. 12 (December), 1990.
- ↑ http://www.sound-ex.com/alternative_zu_soundex
- ↑ <http://www.morfoedro.it/doc.php?n=222&lang=en>
- ↑ B P Pande and Prof. H S Dhami. Article: Application of Natural Language Processing Tools in Stemming. International Journal of Computer Applications 27(6):14-19, August 2011. Published by Foundation of Computer Science, New York, USA.
- ↑ Best Faces Forward: A Large-scale Study of People Search in the Enterprise I Guy, S Ur, I Ronen, S Weber... - 2012 - <http://www.research.ibm.com/haifa/dept/imt/papers/guyCHI12.pdf>
- ↑ <http://aspell.net/metaphone/>
- ↑ <http://www.amorphics.com/>

Categories: [Phonetic algorithms](#)

This page was last modified on 13 July 2015, at 10:21.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

