



new Vista Windows Vista Gadgets competition



Vista

3,903,948 members and growing! 14,619 now online.

Email  Password

[Sign In](#) [Join!](#)

- Home
- MFC/C++
- C#
- ASP.NET
- VB.NET
- Architect
- SQL
- All Topics
- Help!
- Article

All Topics, MFC / C++ >> String >> General (Advanced)

C+  
Wir  
Wir  
Wir  
Dev  
Pos  
Upc  
View

# Implement Phonetic ("Sounds-like") Name Searches with Double Metaphone Part VI: Other Methods & Additional Resources

By [Adam Nelson](#).

Surveys other phonetic matching techniques, and presents additional resources on the subject.

**ANNOUNCEMENTS**

- Vista API competition \$10,000 in prizes
- Vista Gadget comp: \$2,000 in prizes
- Monthly Competition

Search

Articles

[Advanced Search](#)  
[Sitemap](#)

Print Broken Article? Bookmark Discuss Send to a friend

16 votes for [Popula](#)

- Download source files - 111 Kb
- Download demo projects - 616 Kb

## Abstract

Simple information searches -- name lookups, word searches, etc. -- are often implemented using a simple match criterion. However, given both the diversity of homophonic (pronounced the same) words as well as the propensity for humans to misspell surnames, this simplistic criterion often yields results, in the form of reduced result sets, missing records that differ by a misplaced letter or spelling.

This article series discusses Lawrence Phillips' Double Metaphone phonetic matching algorithm and several useful implementations which can be employed in a variety of solutions to create searches of proper names in databases and other collections.

## Introduction

This article series discusses the practical use of the Double Metaphone algorithm to phonetic matching using the author's implementations written for C++, COM ([Visual Basic](#), etc.), scripting languages (ASP), SQL, and .NET (C#, VB.NET, and any other .NET language). For a discussion of the algorithm itself, and Phillips' original code, see Phillips' article in the June 2000 CUJ, available at [http://www.codeproject.com/string/dmetaphone.asp](#).

[Part I](#) introduces Double Metaphone and describes the author's C++ implementation and the use of the author's COM implementation from within Visual Basic. [Part III](#) demonstrates the author's implementation from ASP and with VBScript. [Part IV](#) shows how to perform phonetic matching using the author's extended stored procedure. [Part V](#) demonstrates the author's .NET implementation. [Part VI](#) closes with a survey of phonetic matching alternatives, and pointers to other resources.

## Double Metaphone limitations

While this article series has focused entirely on the Double Metaphone algorithm as a means of phonetic matching, it is worth noting that there are many other algorithms available.

saved on 3/13/2007



new Windows Vista  
**Vista API**  
competition

Write  
a great  
Vista API  
article



phonetic matching in one's applications, Double Metaphone bears some weaknesses that for a particular application, including:

- Though it works as a general-purpose phonetic search algorithm, Double Metaphor works best with, searching lists of proper names rather than large fields of generic
- Double Metaphone provides minimal ranking ability, apart from the three match levels in the series. This limits the ability to tune search results.
- Being a phonetic matching (vs. fuzzy matching like q-grams and edit distances) algorithm, Double Metaphone may fail to match misspelled words when the misspelling substantively changes the structure of a word.

Even bearing these limitations in mind, Double Metaphone is free, efficient, easy to use, and applicable to a wide number of scenarios. Ultimately, only the designer of a particular system can decide if Double Metaphone is appropriate to his/her particular problem space.

## Alternatives to Double Metaphone

Numerous other algorithms and techniques have been developed, each for different purposes and with varying efficacy. This section will explore some of the better-known techniques, and provide information on each method.

### Soundex

Soundex was one of the first, if not the first, formalized phonetic matching algorithm. So widely used by the US Census in the late 19<sup>th</sup> century. Not surprisingly, this algorithm is remarkably inadequate in most cases.

Nonetheless, one encounters Soundex in surprising places, even in modern software solutions. [Microsoft SQL Server](#) offers a **SOUNDEX** function which, given a word, computes Soundex for that word.

For more information on Soundex, a simple Internet search on "soundex" will likely yield many results. Again, the reader is encouraged to consider more advanced alternatives for any product that requires phonetic matching, both primitive and limited.

### Metaphone

Double Metaphone is only the latest incarnation of the Metaphone algorithm, originally published by Philip I. Phillips in 1990. While arguably inferior to Double Metaphone, Metaphone does incorporate some improvements and has the added advantage (and disadvantage) of producing only one phonetic key for a given word.

### Phonix

Phonix is an improved version of Soundex, developed by T.N. Gadd and published in *Association for Computing Machinery's journal, Program* [Gadd, T.N. 'Fishing for words': phonetic retrieval of words in large systems, 22(3) 1988, p. 222] and [Gadd, T.N. PHONIX: the algorithm, 24(4) 1990, p. 36]. Not available online, Phonix has been incorporated into a number of WAIS implementations, including one which is open-source and therefore freely available in source form.

The author has never experimented with Phonix, and therefore cannot write authoritatively about its performance; however being Soundex-based, it bears much of the same baggage which hampers Soundex performance. The paper by Zobel and Dart referenced at the end of this article performs a comparison of Phonix with several other algorithms, producing results which confirm this

### q-Gram based algorithms

A q-gram (sometimes called n-gram, primarily to confuse readers) in this context refers to q letters long, from a given word. For example, for  $q = 2$ , the word **Nelson** has the following

NE EL LS SO ON

By comparison, **Neilsen** breaks down into these q-grams ( $q = 2$ ):

NE EI IL LS SE EN

Clearly, **Nelson** and **Neilsen** share the **NE** and **LS** q-grams in common.

Various techniques have been developed which compare two words based on their q-grams. One would be counting the number of q-grams two words have in common, with a higher count indicating a better match.

Technically, q-gram algorithms aren't strictly phonetic matching, in that they do not operate on the phonetic characteristics of words. Instead, q-grams can be thought of as computing the difference between two words. Since phonetically similar words often have similar spellings, q-grams provide favorable results, yet it also successfully matches misspelled or otherwise mutated words that are rendered phonetically disparate.

### Edit distance based algorithms

Edit distance computes the "distance" between two words by counting the number of insertions, deletions, and substitutions required to permute one word into another. In general the fewer operations required, the better the match. Some implementations assign varying scores to the insert, replace, and delete operations, and common variation varies which operations are considered when computing the distance; some operations may not be considered, thereby defining the edit distance solely in terms of insertions and deletions.

One of the more popular algorithms in the edit distance class is Levenshtein distance (note that some references use the spelling 'Levenstein', which is technically incorrect).

As with q-gram algorithms, edit distance is not strictly phonetic, but often matches words with similarities in spelling.

### Proprietary algorithms

Several organizations offer data scrubbing, de-duplication, data normalizing, and merge-sorting. Some implement some form of approximate text matching, albeit with varying degrees of success. These systems are often proprietary, and seldom documented. The applicability of these systems must be carefully analyzed before adopting any one solution.

### Other sources

The above list of alternative matching algorithms is far from complete, and provides only a beginning search for suitable algorithms -- not to end one. This section lists other sources for phonetic matching, and approximate text matching, as well as links to additional Double Metaphone

## Additional Double Metaphone implementations

- [Links to several Metaphone and Double Metaphone implementations, including C, F](#)

## Additional Phonetic Matching/Approximate text matching resources

- ["Phonetic String Matching: Lessons from Information Retrieval"](#) by Zobel and Dart. comparing all of the matching techniques discussed in this article (except Metaphone unfortunately), and a few more. Includes tables containing quantitative test results
- ["A Guided Tour to Approximate String Matching"](#) by Gonzalo Navarro. - An excellent string matching, which is different from, but related to, phonetic matching.
- ["Approximate Text Searching"](#) by Gonzalo Navarro. - A very exhaustive discussion of matching issues. May be too technical for some readers. Note that the English version is text.
- ["Searching Proper Names in Databases"](#) by Pfeifer, Poersch, and Fuhr. - A very accurate several techniques specifically designed for searching databases of proper (last) names. Includes efficacy tests by the authors.
- ["Learning String Edit Distance"](#) by Ristad, and Yianilos. - Paper describing edit distance function for presenting an interesting stochastic model for learning an edit distance function for the latter subject is likely to be of limited interest to those seeking approximate string matching, however the former topic is highly relevant.
- [Richard Birkby's CodeProject article presenting four Soundex variations in C#](#). - In-depth Soundex performance.
- A very readable [essay by Michael Gillel](#) and describing the Levenshtein Distance algorithm with code in VB, C++, and Java. Be sure to take note of the *Resources* section, which includes links to sites and implementations.
- "Finding String Distances", Dr Dobb's Journal, April 1992, by Ray Valdes. Interesting article on the Levenshtein distance, and its applications not only to string matching but also to handwriting recognition, etc. Article available on DDJ archive CD. Source code available.
- ["A Comparison of String Distance Metrics for Name-Matching Tasks"](#) by Cohen, Raviv, and Eshed. - A nice, technical survey of techniques for measuring the "distance" between strings, with applications to matching proper names. Great source of additional information on edit distance techniques and the Levenshtein distance.
- [SourceForge home for SecondString](#), a Java approximate string matching library by Michael Gillel. Comparison of String Distance Metrics for Name-Matching Tasks" referenced above. Includes things, an implementation of Levenshtein distance.
- [A powerful full-text search engine implemented entirely in Java](#). - Includes Levenshtein distance with all of the typical full text indexing features.
- [AGREP](#) - AGREP is a tool, similar to egrep, fgrep, and grep, which searches for a given pattern in files. AGREP uses an edit distance algorithm to perform approximate matching, making it useful for experimenting with the results one can expect from such algorithms.

## Conclusion

This article concludes the article series on phonetic matching of name data with Double Metaphone. Some of the other major techniques for phonetic matching are presented, as well as links to resources for the reader interested in further research on the subject. By this point, the reader should know that no one solution exists to the problem of matching similar but not identical text, and that care must be taken to select based on the reader's specific criteria.

That said, hopefully this article series will lead the reader to strongly consider Double Metaphone for use, respectable performance, and readily available implementations in a variety of languages.

## History

- 7-22-03 Initial publication
- 7-29-03 Added reference to Richard Birkby's Soundex article
- 7-31-03 Added hyperlinks between articles in the series
- 8-03-03 Added additional resources pertaining to the Levenshtein Distance algorithm

## About Adam Nelson



My name is Adam Nelson. I've been a professional programmer since 1996, working in software development, early first-generation web applications, modern n-tier distributed applications, and security tools, to my last job as a Senior Consultant at BearingPoint posted in Baguetteville, VA. I am currently a tech lead at AppA Virginia startup developing super-secret tools and generally having a lot of fun.

I have a wide range of skills and interests, including cryptography, image processing, military history, 3D graphics, database optimization, and mathematics, to name a few. I also work on a variety of projects (either for my employer or on my own self-edification projects), read, and travel.

Click [here](#) to view Adam Nelson's online profile.

## Other popular String articles:

- [The Complete Guide to C++ Strings, Part II - String Wrapper Classes](#)  
A guide to the string wrapper classes provided by Visual C++ and class libraries
- [The Complete Guide to C++ Strings, Part I - Win32 Character Encodings](#)  
A guide to the multitude of string types used in Windows.
- [CString-clone Using Standard C++](#)  
A Drop-In replacement for CString that builds on the Standard C++ Library's basic\_string template
- [CString Management](#)  
Learn how to effectively use CStrings.



[Top]

Sign in to vote for this article: Poor



**Note:** You must [Sign in](#) to post to this message board.



Message score threshold

View   25

Msgs 1 to 23 of 23 (Total: 23) ([Refresh](#))

Subject

- [SOUNDEX](#)
- [Re: SOUNDEX](#)
- [I can't register the MetaphoneCOM.dll](#)
- [Re: I can't register the MetaphoneCOM.dll](#)

Author

- [RichardGrimm](#)
- [Adam Nelson](#)
- [chuijkh](#)
- [Adam Nelson](#)

saved on 3/13/2007

-  [Re: I can't register the MetaphoneCOM.dll](#)  [chuijkh](#)
-  [Re: I can't register the MetaphoneCOM.dll](#)  [ssteinke](#)
-  [\*\*Has Anybody got this to work?\*\*](#)  [\*\*betterc\*\*](#)
-  [Re: Has Anybody got this to work?](#)  [Adam Nelson](#)
-  [Re: Has Anybody got this to work?](#)  [betterc](#)
-  [Re: Has Anybody got this to work?](#)  [Adam Nelson](#)
-  [Re: Has Anybody got this to work?](#)  [betterc](#)
-  [Re: Has Anybody got this to work?](#)  [Adam Nelson](#)
-  [Re: Has Anybody got this to work?](#)  [betterc](#)
-  [Re: Has Anybody got this to work?](#)  [Adam Nelson](#)
-  [Re: Has Anybody got this to work?](#)  [bug\\_\\_free](#)
-  [Re: Has Anybody got this to work?](#)  [Adam Nelson](#)
-  [I have a working Engine](#)  [shabosco](#)
-  [\*\*Another reference to add to your list\*\*](#)  [\*\*Richard Birkb\*\*](#)
-  [Re: Another reference to add to your list](#)  [Adam Nelson](#)
-  [\*\*Implementation\*\*](#)  [\*\*Rome Singh\*\*](#)
-  [Re: Implementation](#)  [Adam Nelson](#)
-  [Re: Implementation](#)  [Adam Nelson](#)
-  [Re: Implementation](#)  [Rome Singh](#)

Last Visit: 16:14 Tuesday 13th March, 2007

-  [General comment](#)
-  [News / Info](#)
-  [Question](#)
-  [Answer](#)
-  [Joke / Game](#)
-  [Admin message](#)

Updated: 31 Jul 2003

Article conte  
everything else Copyr  
Web08 | [Adverti](#)