Version

1.0

pdSurname User Guide

Last Name Database

A one-of-a-kind proprietary resource designed to facilitate matching last names that are variations or phonetically similar. The package also includes extensive language and race information. The *Pro* edition includes fuzzy logic. Ancestry researchers, students, teachers, and scholars benefit as well because this software is recommended for study in genealogy, onomatology, anthroponymy, ethnology, linguistics, and related disciplines.

Peacock Data, Inc.
California, USA 800-609-9231
Web address: www.peacockdata.com

TABLE OF CONTENTS

INTRODUCTION

Unlike first names, which have been with us since antiquity, fixed surnames were not used much until late in the High Middle Ages when populations grew and people found it necessary to be more specific when talking about someone else. For example, in England, the practice of family names was introduced after the Norman conquest in 1066, but was not fully adopted until the 13th and 14th centuries. The Welsh did not use them until the 17th century, and the Japanese did not have them until the 19th century except among aristocrats. On the other hand, matrilineal surnames existed in China prior to the Shang Dynasty (1600–1046 BC). Ireland was the first country in Europe to adopt fixed last names, "Ó Cleirigh", recorded in 916, being the very first.

In the beginning, surnames were names like John son of Thomas (patronymic), Jane of the Hills (habitational), Henry the Weaver (occupational), and Mary the Redhead (characteristic), until the adoption of modern last names, which were often alterations of these old-fashioned names.

THE PROBLEM

Through this process, many variations of last names occurred, some by design and others due to carelessness or lack of education. For example, as families immigrated to other countries they often modified or even translated their name to fit in with a new language. Many other variations occurred as a largely uneducated society tried to transcribe their names the best way they could while educated families decided to attenuate, accent, or otherwise modify their surnames over time, and Brown becomes Browne. Still other names are not variations at all, but sound similar.

PDSURNAME TO THE RESCUE

Apparently as all this was going on they were not thinking of modern day scribes, typists, and data processers who now need to work with all the variations and phonetic similarities. That is why **pdSurname** was invented. A one-of-a-kind proprietary resource that does for last names what our highly regarded *pdNickname* software does for first names, it is designed to facilitate matching last names that are not exactly the same but are close in relationship, spelling, or sound. Coverage includes surnames from hundreds of languages and the package employs the best matching algorithms designed for this process.

As a further benefit, for a large majority of last names, the language of origin and use have also been researched and included, and all names have a real or estimated calculation for usage among races, including white, black, Hispanic/Latino, Asian/Pacific, Native American/Alaskan, and multiracial use.

An enhanced version even incorporates sophisticated fuzzy logic which allows matching when lists have typographical errors.

This easy-to-use, comprehensive, and up-to-date software is of great value for businesses and organizations working with lists of names, but ancestry researchers, students, teachers, and scholars benefit as well because this software is recommended for study in genealogy, onomatology, anthroponymy, ethnology, linguistics, and related disciplines.

pdSurname is available in **Pro** and **Standard** editions. This guide covers both versions.

PRO EDITION

The *Pro* edition includes more than 80 million standard last name variation records and 28 million fuzzy logic records based on more than 335,000 surname formations, with the relationship identified for name pairs, languages of origin or use, and real or estimated usage among races.

STANDARD EDITION

The *Standard* edition includes the same more than 80 million last name variation records based on more than 335,000 surname formations, with the relationship identified for name pairs, languages of origin or use, and real or estimated usage among races. It has all features of the *Pro* version except the fuzzy logic records.

QUICK START

If you have worked with our *pdNickname* software, you should be comfortable working with *pdSurname*, because they are similar products. The biggest difference is *pdSurname* processes last names while *pdNickname* processes first names.

If you are new to our products, you are in for a wonderful surprise.

pdSurname is a one-of-a-kind proprietary resource designed to facilitate matching last names that are not exactly the same but are close in relationship, spelling, or sound. Coverage includes surnames from hundreds of languages and the package employs the best matching algorithms designed for this process.

This easy-to-use, comprehensive, and up-to-date software is of great value for businesses and organizations working with lists of names, but ancestry researchers, students, teachers, and scholars benefit as well because this software is recommended for study in genealogy, onomatology, anthroponymy, ethnology, linguistics, and related disciplines.

The package is available in *Pro* and *Standard* editions. The only difference is the *Pro* version includes fuzzy logic.

pdSurname is designed to be compatible with any database system. It comes in multiple file formats, uses only the ANSI character set, and has a well-defined layout. It has two main files, a names database and a relationship file.

NAMES DATABASE

The names database includes all the last names in the relationship file along with additional names for which no onomastic or phonetic connection exists.

Each record lists the same surname written in three different ways, stylized, standardized, and normalized. All are in UPPER CASE. **The relationship file is based on the standardized version of the name.** To standardize a name users only need to remove all spaces, periods, hyphens, and apostrophizes.

The race usage of each name is identified in a series of fields which provide an actual or estimated percentage of use for each race, including:

- White (not Hispanic/Latino)
- Black (not Hispanic/Latino)
- Hispanic/Latino
- Asian/Pacific (not Hispanic/Latino)
- Native American/Alaskan (not Hispanic/Latino)
- Multiracial

The language or languages of origin and use are identified in a comma delimited list. If there is more than one language, they are listed alphabetically. None of the languages were derived algorithmically and the provided information represents years of extensive onomastic research.

RELATIONSHIP FILE

The relationship file shows surnames that are connected either onomatologically, phonetically or, in the *Pro* edition only, can be fuzzy logic variations. Onomastic relationships are either close, near, or distant. This is determined by tabulating or estimating the number of lines separating the names on a name tree.

Relationships are flagged as follows:

1 = Close onomastic variant
2 = Near onomastic variant
3 = Distant onomastic variant
P = Phonetic match
F = Fuzzy logic match (*Pro* edition only)

The relationship file is broken into eight (*Standard* edition) or twelve (*Pro* edition) smaller sections for user convenience and to allow users more options when setting up their system.

Each record has two sets of name information, a NAME1 side and a NAME2 side. In one section of the relationship file the name pairs are in alphabetical order and in a second section they are provided with the names reversed.

There are separate sections for records with special characters. Users only need to install the special character section if their lists have extended ANSI characters.

In the *Pro* edition only fuzzy logic file, the setup is different. In one section of the fuzzy file the name pairs are with the fuzzy name first and correct spelling second and in a second section they are provided with the names reversed. The fuzzy logic file also puts names with special characters in a separate section.

The overall quality of each name-pair match is quantified on a scale of 01 (best) to 99. This is because the number of matches from a query can sometimes be very numerous, and the score is effective in ordering the output for filtering. Users will find this a major advantage with our system.

Note that in the *Pro* edition only, a score of 00 is entered for all fuzzy logic matches because they are known to be the same exact name, with one name misspelled.

As part of our phonetic indexing process we include matches from six open source algorithms most data engineers are familiar with. These algorithms are:

- Double Metaphone
- Metaphore
- New York State Identification and Intelligence System (NYSIIS)
- Caverphone
- Soundex
- Daitch–Mokotoff Soundex

Here are some examples of related names:

	Name #1	**Name #2**	**Relation**
Example 1	ACKERMAN	AKERMAN	Close onomastic variant (1)
Example 2	MANCILL	MONSELL	Near onomastic variant (2)
Example 3	WILLIAMSON	WILMSEN	Distant onomastic variant (3)
Example 4	CORREY	CURIE	Phonetic match (P)
Example 5	SANTILLA	SANTOLLA	Phonetic match (P)

FUZZY LOGIC (*PRO* ONLY)

This section applies to pdSurname Pro only.

The fuzzy logic technology in the *Pro* edition of this software allows matching data that has typographical errors. If users look at the fuzzy logic records, they are likely to see errors they have repeatedly made or seen. In many cases you will have to look close to see the difference, but they are different. There are more than 28 million fuzzy logic records.

The fuzzy logic file uses the same format as the other relationship files with a few exceptions. Instead of alphabetical and reverse alphabetical ordered sections, in one section of the fuzzy file the name pairs are with the fuzzy name first and correct spelling second and in a second section they are provided with the names reversed. There is a reason for this. Because the database is so comprehensive, if a name cannot be found in the regular relationship file, particularly on United States lists, it is probably a misspelling. Users should then check the misspellings in the fuzzy logic file.

The most likely typographical errors are determined based on the number of letters, the characters involved, where they are located in the name, the language, and other factors. None of the fuzzy spellings formulate a real name already in the database. This sometimes happens when the fuzzy spelling was already a real variation of the same name.

A score of 00 is entered for all fuzzy logic matches and phonetic algorithms are not run against them because we already known they are the same exact name, with one name misspelled. All fuzzy matches have a relationship of "F".

Some fuzzy logic matches have one typographical error while others have multiple issues, so the technology is suited for even the worst typists and transcribers. The algorithms have five layers:

- Phonetic misspellings
- Reversed digraphs
- Double-letter misspellings
- Missed letters
- String manipulations

*

This quick start explanation demonstrates the basic use the software. Read on about features never before available on this scale.

IMPORTING DATA INTO YOUR SYSTEM

pdSurname is designed to be compatible with any database system. It comes in multiple file formats, uses only the ANSI character set, and has a well-defined layout.

INCLUDED DATABASE FILES

pdSurname has two main files, a names database and a relationship file. The relationship file is broken into sections for user convenience and to allow users more options when setting up their system.

Included files are:

NAMES DATABASE

The names database contains more than 335,000 surname formations. The relationship file is based on these names. For a large majority, the language or languages of origin and use have also been researched and included, and all names have a real or estimated calculation for usage among races, including white, black, Hispanic/Latino, Asian/Pacific, Native American/Alaskan, and multiracial use.

RELATIONSHIP FILE

The relationship file has more than 80 million (*Standard* edition) or 109 million (*Pro* edition) variation records based on the surnames in the names database. It includes the pairs of related names along with extensive information about their onomastic and phonetic relationships. A scoring system is provided to order matches from most likely to least likely.

Matches from our proprietary algorithms are in two sections, one with the name pairs in alphabetical order and the other reversed. Two separate sections are provided with additional matches from six open source phonetic algorithms, one with the name pairs in alphabetical order and the other reversed.

There are separate sections for records with special characters. Users only need to install the special character section if their lists have extended ANSI characters.

In the Pro edition only fuzzy logic file, the setup is different. In one section of the fuzzy file the name pairs are with the fuzzy name first and correct spelling second and in a second section they are provided with the names reversed. The fuzzy logic file also puts names with special characters in a separate section.

FILE FORMATS

The database is available in three common file formats. Each format contains the same data.

Available file formats are:

CSV-COMMA SEPARATED VALUES

Files in Comma Separated Values (CSV) format (also known as Comma Delimited) separate fields with commas, and alpha/numeric character fields are usually delimited with double quotes (in case some of the field content includes commas). This format is the most commonly used. It is a native format for Microsoft Excel and is compatible with nearly all database management systems and spreadsheets.

TXT-FIXED LENGTH

Files in Fixed Length (TXT) format (also known as Standard Data Format or SDF) use constant field positions and lengths for all records. In other words, each field starts and ends at the same place in the text file and each record is on a separate line. While not as popular as comma separated values, this format is preferred by many due to its input precision and is widely used to transfer data between different software programs. It is compatible with most database management systems and spreadsheets.

DBF-DATABASE

Files in DBF database format (also known as xBase) are native to Microsoft FoxPro and Visual FoxPro, dataBased Intelligence dBase, Alaska Software XBase++, Apollo Database Engine, Apycom Software DBFView, Astersoft DBF Manager, DS-Datasoft Visual DBU, Elsoft DBF Commander, GrafX Software Clipper and Vulcan.NET, Multisoft FlagShip, Recital Software Recital, Software Perspectives Cule.Net, and xHarbour.com xHarbour. They are also compatible with any database management system that can import the DBF (xBase) format, such as Microsoft Access, Microsoft SQL Server, and numerous others.

CHARACTER SET

The ANSI character set is utilized for all database records. This includes ASCII values 0 to 127 and extended values 128 to 255. These are also known as the extended Latin alphabet. Some users may needs to configure their database system to import the extended values. In many cases the option will be labeled the “Latin-1” character set.

FILE LAYOUTS AND DATA DEFINITIONS

Below are the complete layout specifications and data definitions of all files provided with *pdSurname*.

Each line below contains the following information: FIELD NUMBER: field position number. FIELD NAME: name of field. FIELD TYPE: field data type; "*Chr*" = alpha/numeric characters, "Num" = numbers. FIELD LENGTH: length of field. DECIMAL PLACES: number of decimal places (if any). START POSITION: field starting position. END POSITION: field ending position. DESCRIPTION: data definition of field contents.

LAYOUT OF PDSURNAME NAMES DATABASE

Field Count: 13

Total Length: 418

Record Count: Pro and Standard: 336,955

FIELD NUMBER	FIELD NAME	FIELD TYPE	FIELD LENGTH	DECIMAL PLACES	START POSITION	END POSITION	DESCRIPTION
1	**PEACOCK_ID**	Chr	16		1	16	**Unique identifier for each record**
2	**NORMAL**	Chr	35		17	51	**Normalized surname spelling**
3	**STANDARD**	Chr	35		52	86	**Standardized surname spelling**
4	**NAME**	Num	35		87	121	**Stylized surname spelling**
5	**WHITE**	Num	6	2	122	127	**Percent white (not Hispanic/Latino)**
6	**BLACK**	Num	6	2	128	133	**Percent black (not Hispanic/Latino)**
7	**HISPANIC**	Num	6	2	134	139	**Percent Hispanic/Latino**
8	**ASIAN**	Num	6	2	140	145	**Percent Asian/Pacific (not Hispanic/Latino)**
9	**NATIVE**	Num	6	2	146	151	**Percent Native American/Alaskan (not Hispanic/Latino)**
10	**MULTIRACE**	Num	6	2	152	157	**Percent multiracial**
11	**RACECONF**	Chr	1		158	158	**Race confidence flag:** *1 = Tabulation (not an estimate)* *2 = Estimate—high confidence* *3 = Estimate—medium confidence* *4 = Estimate—low confidence*
12	**RANK**	Num	6	0	159	164	**Name rank from 2000 U.S. Census**
13	**LANGUAGE**	Chr	254		165	418	**Language or languages of origin and use**

LAYOUT OF PDSURNAME RELATIONSHIP FILE

Field Count: 11

Total Length: 80

Record Count: Pro: 109,595,846; Standard: 80,742,846

FIELD NUMBER	FIELD NAME	FIELD TYPE	FIELD LENGTH	DECIMAL PLACES	START POSITION	END POSITION	DESCRIPTION
1	**NAME1**	Chr	35		1	35	**First surname in the related name pair**
2	**NAME2**	Chr	35		36	70	**Second surname in the related name pair**
3	**REL**	Chr	1		71	71	**Relationship:** *1 = Close onomastic variant* *2 = Near onomastic variant* *3 = Distant onomastic variant* *P = Phonetic match* *F = Fuzzy logic match* (Pro *only)* *Blank = Only matched open source algorithms; possibly a false match*
4	**SCORE**	Chr	2		72	73	**Match quality score:** *01 (best) to 99 score; fuzzy logic matches receive a score of 00*
5	**DMP**	Chr	1		74	74	**Double Metaphore:** *P = Primary line match* *S = Secondary line match*
6	**MP**	Chr	1		75	75	**Metaphone** *M = match*
7	**NY**	Chr	1		76	76	**New York State Identification and Intelligence System (NYSIIS):** *N = match*
8	**CP**	Chr	1		77	77	**Caverphone:** *C = match*
9	**SX**	Chr	1		78	78	**Soundex:** *S = match*
10	**DMSX**	Chr	1		79	79	**Daitch–Mokotoff Soundex:** *P = Primary line match* *S = Secondary line match*
11	**ABC**	Chr	1		80	80	**Alphabetical order flag:** *A = Name pair is in alphabetical order (or, in the Pro edition only fuzzy logic file, fuzzy name first and correct spelling second)* *R = Names pair is in reverse alphabetical order (or, in the Pro edition only fuzzy logic file, fuzzy name first and correct spelling second)*

SPECIAL FEATURES

We try to make all our software great, but during each development cycle certain aspects are emphasized. In this version of *pdSurname* special attention was paid to the following:

- **Hispanic/Latino names**: including Spanish, Basque, Catalan, Galician, and Portuguese
- **Native American names**: in fact we designed this to be the definitive collection
- **Irish names**: due to the excellent records maintained by the National Library of Ireland
- **English names**: including Anglo-Saxon, Middle English, and Modern English
- **Ashkenazi Jewish names**: particularly Polish Jewish, German Jewish, and Czech Jewish
- **Prefix names**: our algorithms are specially tuned to work effectively with names that have prefixes such as "MC", "MAC", "O", "DE", "LA", "VAN", "AL", "ST", and many others

USING THE NAMES DATABASE

***pdSurname* has two databases, a names database with all the surnames, languages, and race information, and another with name-pair relationships. This section discusses the names database.**

The names database includes all the last names in the relationship file along with additional names for which no onomastic or phonetic connection exists. For a large majority, the language of origin and use have also been researched and included, and all names have a real or estimated calculation for usage among races, including white, black, Hispanic/Latino, Asian/Pacific, Native American/Alaskan, and multiracial use.

PEACOCK_ID FIELD

FIELDS

PEACOCK_ID | Unique identification number (primary key)
Each record has a 16-character alphabetic primary key that uniquely distinguishes it from all other records in the table.

The first field in the names database is PEACOCK_ID. It provides a unique primary key identifier for each record. Each begins with the character "s" to identify the product. Each identification number has three parts and each part is separated with a hyphen.

The following is the first PEACOCK_ID in the names database:

- **s0000001-001-001** is a complete PEACOCK_ID; no other record has this same exact identification

PARTS OF PEACOCK_ID

The PEACOCK_ID primary key is made up of the following three parts:

s0000001 is the first part of a PEACOCK_ID; it identifies each unique normalized name; multiple records can have this same number but each record is showing the same normalized name with other formations of the name.

s0000001-001 are the first and second parts of a PEACOCK_ID; they identify each standardized name formation with the same normalized name; multiple records can have this same number but each record is showing the same standardized name with other formations of the name.

s0000001-001-001 are all three parts of a PEACOCK_ID; they identify each stylized name formation with the same standardized and normalized names; multiple records cannot have this same number.

NAME FIELDS

FIELDS

NORMAL | Normalized surname spelling
STANDARD | Standardized surname spelling
NAME | Stylized surname spelling
Each record has three up to 35-character alphabetic names that indicated the normalized, standardized, and stylized spelling of the same name. Different styling can indicate different languages and race information. All names are in UPPER CASE.

Each record lists the same surname written in three different ways, stylized, standardized, and normalized. All are in UPPER CASE. **The relationship file is based on the standardized version of the name.**

Note that a great majority of names do not contain any styling or special characters so all three versions of the name will be the same.

See the end of this section for examples of stylized, standardized, and normalized names. Note when the stylized, standardized, and normalized names are all different, all the same, or when the stylized name is different but the standardized and normalized names are the same.

NAME (STYLIZED)

These are last names in formations commonly found in name lists with any styling intact. Names sometimes have hyphens, periods, spaces, accents, and other special characters. Users do not need to remove any styling or special characters when searching on this field. When a name exists in multiple formations, it is grouped and listed separately for each stylization. Different styling and the use of special characters can indicate different languages and race information. All names are in UPPER CASE.

The stylized formations are from lists encountered during years of extensive onomastic research. For this reason users will find more stylizations for last names that are more common in various nationalities and when onomastic information is more abundant. Additional theoretical formations are sometimes possible.

STANDARDIZED

These are the same as stylized last name formations, but with all spaces, periods, hyphens, and apostrophizes removed. **This formation is used to build and compare the names in the relationship file.** Because names can be written in a number of ways, removing these characters makes the name data easier to match. Most database systems have simple commands to accomplish this. Most English and Americanized names do not require standardization. Standardization transforms the following characters:

From	To	Description
-	*	Hyphen
.	*	Period

* = Removed

From	To	Description
‘	*	Left single quote/apostrophe
’	*	Right single quote/apostrophe

* = Removed

From	To	Description
'	*	Apostrophe
	*	Blank space

* = Removed

NORMALIZED

These are the same as standardized last name formations, but with all non-English alphabetic letters and glyphs converted English A—Z letters, such as grave accents, acute accents, umlauts, and other values in the extended ANSI character set. Additionally, names starting with a "SAINT" or "STE" prefix are converted to "ST". **Users do not need to normalize name information to match against the relationship file, but the provided special character database will need to be installed if lists have extended ANSI characters.** Most English and Americanized names do not require normalization. Normalization transforms the following characters:

From	To	Description
À	A	A-grave
Á	A	A-acute
Â	A	A-circumflex
Ã	A	A-tilde
Ä	A	A-diaeresis (umlaut)
Å	A	A-ring
Æ	AE	Æsc (grapheme)
Ç	C	C-cedilla
Ð	D	Eth
È	E	E-grave
É	E	E-acute
Ê	E	E-circumflex
Ë	E	E-diaeresis (umlaut)

From	To	Description
Ì	I	I-grave
Í	I	I-acute
Î	I	I-circumflex
Ï	I	I-diaeresis (umlaut)
Ñ	N	N-tilde
Ò	O	O-grave
Ó	O	O-acute
Ô	O	O-circumflex
Õ	O	O-tilde
Ö	O	O-diaeresis (umlaut)
Ø	O	Ø-vowel
Œ	OE	Œ (grapheme)
Š	S	S-caron (grapheme)

From	To	Description
ß	SS	Eszett/Sharp S
Ù	U	U-grave
Ú	U	U-acute
Û	U	U-circumflex
Ü	U	U-diaeresis (umlaut)
Ý	Y	Y-acute
Þ	Y	Þorn (Thorn)[Þ]
Ž	Z	Z-caron (grapheme)
¸	*	Spaced cedilla
`	*	Spaced grave accent
´	*	Spaced acute accent
SAINT**	ST	SAINT-prefix
STE**	ST	STE-prefix

* = Removed

** = The SAINT and STE transformations should only be performed when they are being used as prefixes (when followed by a period, hyphen, or space). Normally these prefixes are French, but can be English and German among other sporadic use.

[Þ] The Þorn (Thorn) has a different use today than in the days of Old English. It is now used to make the "Y" sound in "Ye Olde". In actuality, in the days of the Anglo-Saxons it was used to pronounce what the Norman French would later introduce as the digraph "TH". They did not pronounce it with a "Y' sound. After the invention of the printing press, parts of the Þorn were abbreviated or dropped to the point it resembled a "Y". That is the story of how we came to mispronounce Old English. Note that "Ye" is still used informally in Hiberno-English (Irish English).

Note that an exception to substituting extended ANSI characters are three one-letter Minnan Chinese names ("Î", "Ô", and "Û") which are not transformed.

EXAMPLES

The following are examples of stylized, standardized, and normalized names. Note when the stylized, standardized, and normalized names are all different, all the same, or when the stylized name is different but the standardized and normalized names the same:

Normalized	Standardized	Stylized	Language
ALCHAHUAN	ALCHAHUÁN	AL-CHAHUÁN	Arabic
CARBAJAL	CARBAJAL	CARBAJAL	Spanish
DEGARCIA	DEGARCIA	DE GARCIA	Catalan and Spanish
MCCNAIMHIN	MCCNÁIMHÍN	MC CNÁIMHÍN	Irish
OSHANNESSY	OSHANNESSY	O'SHANNESSY	Irish
OSBORNE	OSBORNE	OSBORNE	English
STPETER	SAINTPETER	SAINT PETER	Anglicized French
STPIERRE	STPIERRE	ST. PIERRE	French

RACE

FIELDS

WHITE | White race (not Hispanic/Latino)
BLACK | Black race (not Hispanic/Latino)
HISPANIC | Hispanic/Latino
ASIAN | Asian or Pacific Island race
NATIVE | Native American or Alaskan race (not Hispanic/Latino)
MULTIRACE | Multiple races
Each record has six numeric fields with up to three-digits and two decimal places (0.00 to 100.00) that indicates as a percentage the racial usage of the surname in six racial groups.

RACECONF | Race confidence flag
Each record has a one-character alphabetic code that indicates the confidence in the race information:
1 = Tabulation (not an estimate)
2 = Estimate—high confidence
3 = Estimate—medium confidence
4 = Estimate—low confidence

The race usage of each name is identified in a series of fields which provide an actual or estimated percentage of use for each race, including:

- White (not Hispanic/Latino)
- Black (not Hispanic/Latino)
- Hispanic/Latino
- Asian/Pacific (not Hispanic/Latino)
- Native American/Alaskan (not Hispanic/Latino)
- Multiracial

The race applies to the stylized name. Differently styled names can have different race values.

IMPORTANT

The ratios have distinct biases. For names in the United States, the usage is provided from U.S. Census Bureau documents derived from their year 2000 decimal national tabulation. For international names, either similar international information was utilized or estimates were derived by comparing similar names in the same language. For estimated race percentages, a confidence level is indicated, from 2 (high confidence) to 4. A confidence of 1 is entered if it is a tabulation and not an estimate. Because so little reliable information is available for names with special characters, such as accents, as a matter of course, the highest rating given to these names is 3. Confidence levels are not provided for languages because none were derived algorithmically.

It is also important to understand that the definition of race for the U.S. Census Bureau is whatever race respondents identify with. For example, while Portuguese is in the same language family as Spanish (West Iberian), many Portuguese people do not consider themselves Hispanic/Latino. In a similar manor, while a large number of Arabic and Persian speakers live in Western Asia, they may identify as white and not Asian. For this reason the language information may be very important for some projects.

RANK

FIELDS

RANK | Census name rank
Records have an up six digit numeric value indicating name rank from the 2000 U.S. Census; names not in the census list do not have ranking.

The U.S. Census Bureau published a list of surnames occurring 100 times or more from the 2000 decimal census. The list contains 151,671 surnames, and we have provided the ranking for each name included in names database. Note that *pdSurname* includes many names not in the census list, so these names do not have ranking. The most popular name is “Smith” with 2,376,207 entries or 880.85 frequency per 100,000. The most popular Hispanic/Latino name is “Garcia” ranked #8 with 858,289 entries or 318.17 frequency per 100,000.

LANGUAGE

FIELDS

LANGUAGE | Language string
Each record with language information has an up to 254-character alphabetic list that indicates the language or languages of use of the surname. Multiple languages are entered as a comma-delimited list with the languages in alphabetical order.

The language or languages of origin and use are identified in a comma delimited list. If there is more than one language, they are listed alphabetically. None of the languages were derived algorithmically and the provided information represents years of extensive onomastic research. When different sources list different origins and usages they may be combined depending on the reliability of the source and the reasonability of the information. The languages apply to the stylized name. Differently styled names can have different language values.

Language coverage is extensive. The list exceeds 600 languages, language families, and dialects. Some languages refer to ethnic groups.

TOP 30 LANGUAGES

These following are the top 30 languages with the number of occurrences in the names database. The language count is one for each unique name formation and not one for each relationship (which would be many more):

1.	Polish Jewish	36,900	11.	Other Jewish	7,700	21.	Hindi	2,000
2.	Irish	35,500	12.	Dutch	7,100	22.	Hungarian	1,700
3.	Czech Jewish	30,200	13.	Russian	5,400	23.	Swedish	1,700
4.	German Jewish	22,500	14.	Polish	5,000	24.	Middle English	1,600
5.	German	20,200	15.	Catalan	3,500	25.	Norwegian	1,400
6.	Spanish	16,700	16.	Armenian	2,800	26.	Indian	1,300
7.	French	14,600	17.	Arabic	2,500	27.	Turkish	1,300
8.	English	14,500	18.	Native Dakota	2,500	28.	Anglicized Irish	1,200
9.	Italian	12,800	19.	Japanese	2,200	29.	Ukrainian	1,100
10.	Scottish	12,400	20.	Czech	2,200	30.	Welsh	1,100

Note that the counts are rounded to the lower 100.

USING THE RELATIONSHIP FILE

***pdSurname* has two databases, a names database with all the surnames, languages, and race information, and another with name-pair relationships. This section discusses the relationship file.**

The relationship file shows surnames that are connected either onomatologically, phonetically or, in the *Pro* edition only, can be fuzzy logic variations. Onomastic variations are either close, near, or distant.

Here are some examples of related names:

	Name #1	**Name #2**	**Relation**
Example 1	ACKERMAN	AKERMAN	Close onomastic variant (1)
Example 2	MANCILL	MONSELL	Near onomastic variant (2)
Example 3	WILLIAMSON	WILMSEN	Distant onomastic variant (3)
Example 4	CORREY	CURIE	Phonetic match (P)
Example 5	SANTILLA	SANTOLLA	Phonetic match (P)

Our algorithms are specially tuned to work effectively with names that have prefixes such as "MC", "MAC", "O", "DE", "LA", "VAN", "AL", "ST", and many others. Traditionally phonetic algorithms have difficulty with these names because the prefixes create numerous false matches and miss true matches. Our algorithm greatly reduces this problem by separately measuring both the full name and the main part of the name following the prefix. Knowing the language of the name is key to our technique. Users will find this a major advantage with our system. Here are examples:

	Name #1	**Name #2**	**Notes**
Example 6	MCARTHUR	MCDALE	FALSE MATCH: Matched by open sources but not by our proprietary algorithms
Example 7	DEGARCIA	GARCIA	TRUE MATCH: Matched only by our proprietary algorithms

The relationship file is broken into eight (*Standard* edition) or twelve (*Pro* edition) smaller sections for user convenience and to allow users more options when setting up their system. Sections include:

1. Variations and phonetic name pairs in alphabetical order
2. Variations and phonetic name pairs in reverse alphabetical order
3. Open source-only phonetic matches in alphabetical order*
4. Open source-only phonetic matches in reverse alphabetical order*
5. Special character name pairs in alphabetical order**
6. Special character name pairs in reverse alphabetical order**
7. Special character open source-only name pairs in alphabetical order**
8. Special character open source-only name pairs in reverse alphabetical order**
9. PRO ONLY: Fuzzy logic name pairs with the fuzzy name first and correct spelling second
10. PRO ONLY: Fuzzy logic name pairs with the correct spelling first and fuzzy name second
11. PRO ONLY: Special character fuzzy logic name pairs with the fuzzy name first and correct spelling second**
12. PRO ONLY: Special character fuzzy logic name pairs with the correct spelling first and fuzzy name second**

** If an open source phonetic match is also in a primary variation file, the name pair is provided in the primary file only and not repeated in both. All open source matches are individually flagged.*

***Relationships where at least one name has an extended ANSI character*

Matches from our proprietary algorithms are in two sections, one with the name pairs in alphabetical order and the other reversed. Two separate sections are provided with additional matches from six open source phonetic algorithms, one with the name pairs in alphabetical order and the other reversed.

There are separate sections for records with special characters. Users only need to install the special character section if their lists have extended ANSI characters.

In the Pro edition only fuzzy logic file, the setup is different. In one section of the fuzzy file the name pairs are with the fuzzy name first and correct spelling second and in a second section they are provided with the names reversed. The fuzzy logic file also puts names with special characters in a separate section.

All surnames in the relationship file are drawn from the standardized version in the names database. Standardizing requires all spaces, periods, hyphens, and apostrophizes be removed. Users do not need to normalize name information (convert extended ANSI characters, such as accents, to English characters), but the separate special character relationship file will need to be installed if lists have extended ANSI characters. If users do normalize their names they will still match against the regular relationship file.

Note that for the most part the names in the open source files are lower level matches. Anyone who has used any of the open source phonetic algorithms knows that they can deliver a lot of false matches. An important part of our algorithm filters most the “junk” into the secondary files. No algorithm is perfect, however, and users should not totally ignore the open source files. Also note that to prevent too many good matches from being put into the secondary files, there will be some false matches in the main files, but they will usually be sorted to the bottom of the list.

NAME FIELDS

FIELDS

NAME1 | Name #1
NAME2 | Name #2
Each record has a pair of related names in fields that can be up to 35 alphabetic characters each.

Each record has two sets of name information, a NAME1 side and a NAME2 side. The surnames are related either onomatologically, phonetically or, in the *Pro* edition only, can be fuzzy logic variations. In one section of the relationship file the name pairs are in alphabetical order and in a second section they are provided with the names reversed.

There are separate sections for records with special characters. Users only need to install the special character section if their lists have extended ANSI characters.

In the *Pro* edition only fuzzy logic file, the setup is different. In one section of the fuzzy file the name pairs are with the fuzzy name first and correct spelling second and in a second section they are provided with the names reversed. The fuzzy logic file also puts names with special characters in a separate section.

RELATIONSHIP FLAG

FIELDS

REL | Relationship flag
Each record has a one-character alphabetic code that indicates the relationship between name pairs:
1 = Close onomastic variant
2 = Near onomastic variant
3 = Distant onomastic variant
P = Phonetic match
F = Fuzzy logic match (*Pro* edition only)
Blank = Only matched open source algorithms; possibly a false match

The relationship between each name pair is identified a close, near, or distant onomastic variation, as a phonetic variation or, in the *Pro* edition only, as a fuzzy logic variation. In some cases the relationship information is blank because our proprietary algorithms filtered it out as a false record but it was matched by one or more of the open source phonetic algorithms. All of the blank relationships are in the open source relationship files so they can be easily filtered.

The onomastic distance of true variations is rated on a 1 (closest) to 3 scale. The value is determined by tabulating or estimating the number of lines separating the names on a name tree.

SCORE

FIELDS

SCORE | Match quality score
Each record has a two-character numeric code that indicates on a 01 (best) to 99 scale quantifying the quality of each name-pair match. In the Pro *edition only, a score of 00 is entered for all fuzzy logic matches because they are known to be the same exact name, with one name misspelled.*

The overall quality of each name-pair match is quantified on a scale of 01 (best) to 99. The scoring considers several factors:

- Phonetic points from our proprietary algorithm
- How many open source algorithms were matched
- How close the languages and race match
- If the name pairs are onomatologically linked

The number of matches from a query can sometimes be very numerous, and the score is effective in ordering the output for filtering. Users will find this a major advantage with our system.

Note that in the *Pro* edition only, a score of 00 is entered for all fuzzy logic matches because they are known to be the same exact name, with one name misspelled.

OPEN SOURCE PHONETIC ALGORITHMS

FIELDS

DMP | Double Metaphone | *P = Primary line match; S = Secondary line match*
MP | Metaphone | *M = match*
NY | New York State Identification and Intelligence System (NYSIIS) | *N = match*
CV | Caverphone | *C = match*
SX | Soundex | *S = match*
DMSX | Daitch–Mokotoff Soundex | *P = Primary line match; S = Secondary line match*
Each record has up to six one-character alphabetic codes that indicates if a particular open source phonetic match was achieved. Flags are indicated above.

As part of our phonetic indexing process we include matches from six open source algorithms most data engineers are familiar with. These algorithms are:

SOUNDEX

This is the original phonetic algorithm. It was developed by Robert C. Russell and Margaret King Odell and patented in 1918 and 1922. The process was the first to index names by sound, as pronounced in English. The algorithm mainly encodes consonants. A vowel is not encoded unless it is the first letter.

METAPHONE

This is considered the first advanced phonetic algorithm. It was published in 1990 by Lawrence Philips and improved on Soundex by using information about variations and inconsistencies in English spelling and pronunciation to produce more accurate coding.

DOUBLE METAPHONE

This algorithm, also published by Lawrence Philips, is called “Double” because it can return both a primary and a secondary code for a name string. The algorithm takes into account spelling peculiarities of a number of languages in addition to English.

NEW YORK STATE IDENTIFICATION AND INTELLIGENCE SYSTEM (NYSIIS)

This algorithm was developed in 1970 and is similar to Soundex except it maintains relative vowel positioning and handles some phonemes and sequential letters better. The accuracy increase over Soundex has been cited as 2.7 percent.

CAVERPHONE

This algorithm was first developed by David Hood in the Caversham Project at the University of Otago in New Zealand in 2002 and revised in 2004. It was created to assist in data matching between late 19th century and early 20th century New Zealand electoral rolls.

DAITCH–MOKOTOFF SOUNDEX

This algorithm was developed in 1985 by Jewish genealogists Gary Mokotoff and Randy Daitch. It is a refinement of Soundex algorithms designed to allow greater accuracy in matching of Eastern European and Ashkenazi Jewish surnames with similar pronunciation but differences in spelling.

FUZZY LOGIC

This section applies to pdSurname Pro only.

The fuzzy logic technology in the *Pro* edition of this software allows matching data that has typographical errors. If users look at the fuzzy logic records, they are likely to see errors they have repeatedly made or seen. In many cases you will have to look close to see the difference, but they are different. There are more than 28 million fuzzy logic records.

The fuzzy logic file uses the same format as the other relationship files with a few exceptions. Instead of alphabetical and reverse alphabetical ordered sections, in one section of the fuzzy file the name pairs are with the fuzzy name first and correct spelling second and in a second section they are provided with the names reversed. There is a reason for this. Because the database is so comprehensive, if a name cannot be found in the regular relationship file, particularly on United States lists, it is probably a misspelling. Users should then check the misspellings in the fuzzy logic file. The following four sections are provided:

1. Fuzzy logic name pairs with the fuzzy name first and correct spelling second
2. Fuzzy logic name pairs with the correct spelling first and fuzzy name second
3. Special character fuzzy logic name pairs with the fuzzy name first and correct spelling second*
4. Special character fuzzy logic name pairs with the correct spelling first and fuzzy name second*

**Relationships where at least one name has an extended ANSI character*

The most likely typographical errors are determined based on the number of letters, the characters involved, where they are located in the name, the language, and other factors. None of the fuzzy spellings formulate a real name already in the database. This sometimes happens when the fuzzy spelling was already a real variation of the same name.

A score of 00 is entered for all fuzzy logic matches and phonetic algorithms are not run against them because we already known they are the same exact name, with one name misspelled. All fuzzy matches have a relationship of "F" and there is a separate section for records with special characters. Users only need to install the special character section if their lists have extended ANSI characters.

Some fuzzy logic matches have one typographical error while others have multiple issues, so the technology is suited for even the worst typists and transcribers. The algorithms have five layers:

PHONETIC MISSPELLINGS

These algorithms look at digraphs, trigraphs, tetragraphs, pentagraphs, hexagraphs, and even a German heptagraph, "SCHTSCH", used to translate Russian words with the "SHCHA" or "SHCH" (romanticized) sound. These are, respectively, two to seven letter sequences that form one phoneme or distinct sound. Most of letter sequences trigraph and above are Irish who have more language rules than you can shake a stick at.

Many misspellings occur as transcribers enter the sounds they hear. The character sequences and the sounds they produce are different for each language and situation, such as before, after, or between certain vowels and consonants, so our substitutions are language-rule based. Furthermore, our algorithms consider both how a name

may sound to someone who speaks English as well as how it may sound to someone who speaks Spanish, which is often different. Take the digraph "SC". Before the vowels "E" or "I" it is most likely to be misspelled by an English speaker as "SHE" or "SHI" while a Spanish speaker may hear "CHE" or "CHI" and sometimes "YE" or "YI". Our library includes over 80,000 language-based letter sequence phonetic rules. Phonetic misspelling examples:

	Fuzzy name	**Real name**
Example 1	ALLANO	AGLIANO
Example 2	GUALTIEREZ	GUALTIERREZ
Example 3	HEATHFALD	HEATHFIELD
Example 4	OUGHGARD	AAGARD
Example 5	YONGMAN	YOUNGMAN

REVERSED DIGRAPHS

These algorithms look for misspellings due to reversed digraphs (two letter sequences that form one phoneme or distinct sound) which are a common typographical issue, such as "IE" substituted for "EI". The character sequences and the sounds they produce are different for each language and situation, such as before, after, or between certain vowels and consonants, so our substitutions are language-rule based. Reversed digraph examples:

	Fuzzy name	**Real name**
Example 6	ANLGES	ANGLES
Example 7	DEILEMAN	DIELEMAN
Example 8	OLAERY	OLEARY
Example 9	RODREUGEZ	RODREGUEZ
Example 10	SCHUMAHCER	SCHUMACHER

DOUBLE-LETTER MISSPELLINGS

These algorithms look for misspellings due to double letters typed as single letters and single letters that are doubled. The most common typographical issues occur with the characters, in order of frequency, "SS", "EE", "TT", "FF", "LL", "MM", and "OO". Double-letter misspelling examples:

	Fuzzy name	**Real name**
Example 11	HUMBEER	HUMBER
Example 12	ZWOLE	ZWOLLE

MISSED LETTERS

These algorithms look for missed keystrokes and provide fuzzy logic matches with missing letters. Unlike the other algorithms, these are not language specific. Keystrokes can be missed in any language. Missed letter examples:

	Fuzzy name	**Real name**
Example 13	UNTER	HUNTER
Example 14	TAMRON	TAMERON

STRING MANIPULATION

Because so many of our algorithms are language-rule bases, additional name string manipulations are provided for the relatively small number of names without language applied. Most of these are similar to the reversed digraph substitutions. String manipulation examples:

	Fuzzy name	**Real name**
Example 15	ELWROTHY	ELWORTHY
Example 16	POEPLE	PEOPLE

REVERSE RECORDS

FIELDS

ABC | Alphabetic order flag

Each record has one-character alphabetic codes that indicates if a name pair is in alphabetical order or reversed or, in the Pro *edition only fuzzy logic file, if it has the fuzzy spelling or correct spelling first:*

A = Name pair is in alphabetical order (or, in the *Pro* edition only fuzzy logic file, fuzzy name first and correct spelling second)

R = Name pair is in reverse alphabetical order (or, in the *Pro* edition only fuzzy logic file, fuzzy name first and correct spelling second)

In one section of the relationship file the name pairs are in alphabetical order and in a second section they are provided with the names reversed.

In the Pro edition only fuzzy logic file, the setup is different. In one section of the fuzzy file the name pairs are with the fuzzy name first and correct spelling second and in a second section they are provided with the names reversed.

COMPATIBILITY

To ensure compatibility with any operating system and database platform, *pdSurname* is provided in multiple file formats and utilizes only the ANSI character set (ASCII values 0 to 127 and extended values 128 to 255).

USING PDSURNAME WITH PDNICKNAME AND PDGENDER

pdSurname, pdNickname, and *pdGender* make excellent partners. They have been developed to be fully compatible. The name pair format in *pdSurname* is very similar to the *pdNickname* database except *pdSurname* is used to match last names while *pdNickname* matches first names. *pdGender* is based on the first name database and is designed to apply gender identification to first name records. Note that *pdNickname* and *pdGender* are not required to use *pdSurname* but they are highly attuned to work together.

USER GUIDE UPDATES

User guides are updated based on information gained from user experience. It is suggested that users regularly check the Support section of the Peacock Data website for updates. Look for a date newer than the date below:

The publication date of this guide is: March 2, 2015.

DATABASE VERSION NUMBER

Depending on the file format, the version number of each copy of *pdSurname* is written in the first or second row of the first or second column of all database files in **X.X.X** format. The first number is the main version number of the release. The number after the first dot is the update for the version indicated. The number after the second dot references a minor revision.

SITE LICENSE

Peacock Data's site licenses are designed to be fair. They are broader than most software licenses in that they allow installation on not one but all computers in the same building within a single company or organization. We ask users to honor these simple rules so Peacock Data can continue bringing great products to users.

THE USE OF *PDSURNAME* IS GOVERNED BY THE FOLLOWING SITE LICENSE

I. This Site License grants to the Licensee the right to install the licensed version of **pdSurname** (hereinafter, 'information') on all computers in the same building within a single company or organization. Separate Site Licenses must be purchased for each building the information is used in.

II. The information may only be used by the employees of the Licensee. If the Licensee is an educational institution, the data may only be used by enrolled students, faculty, teaching assistants, and administrators.

III. Temporary employees, contractors, and consultants of the Licensee who work on-site at the Licensee's facility may also use the information in connection with the operation of the business of the Licensee. Any copies of the information used by temporary employees, contractors, and consultants must be removed from such individual's computers once they cease working at the Licensee's facility.

IV. The information cannot be used to provide services or products to customers or other third parties, whether for-profit or given away. A Developer License must be purchased separately by the Licensee to incorporate the information in for-profit services and products.

V. The Licensee is required to use commercially reasonable efforts to protect the information and restrict network or any other access to the information by anyone inside or outside of the Licensee's facility who is not authorized to use the information.

VI. The Licensee owns the media on which the information is recorded or fixed, but the Licensee acknowledges that Peacock Data, Inc. and its licensors retain ownership of the information itself.

VII. The Licensee may not transfer or assign its rights under this license to another party without Peacock Data, Inc.'s prior written consent.

VIII. Peacock Data, Inc. may revoke the rights granted by this license upon a violation of any provision herein by the Licensee.

IX. This Site License is governed by Peacock Data, Inc.'s Terms of Service and Privacy Policy, and the laws and regulations of the United States and the State of California.

COPYRIGHT NOTICE