

The nameauth package

Charles P. Schaum
charles dot schaum at att dot net

v1.4 from 2012/07/24

Abstract

The `nameauth` package automates the formatting and indexing of names, facilitating the implementation of a **name authority**. This allows one to move blocks of text without retyping names, making it easier to go from drafts to a final manuscript. This package mainly supports Western names, with basic features for ancient, royal, and Eastern names.

Contents

1	Introduction	2	2.4.5	Name Formatting .	12
1.1	Design Notes and Thanks	2	2.4.6	Formatting Certain Sections . . .	13
1.2	Disclaimer	2	2.4.7	Tweaks: <code>\ForgetName</code> and <code>\SubvertName</code>	14
1.3	Caveat	2	2.5	Name Variant Macros . .	14
2	Usage	3	2.5.1	Introduction . . .	14
2.1	Package Options	3	2.5.2	<code>\AKA</code>	15
2.2	Quick Start Guide	4	2.5.3	<code>\PName</code>	16
2.2.1	Hints for Easier Usage	4	2.6	Indexing Macros	16
2.2.2	Error Handling . .	6	2.6.1	<code>\IndexName</code>	16
2.3	Naming Macros	7	2.6.2	<code>\TagName</code>	17
2.3.1	Surnames: <code>\Name</code> and <code>\Name*</code>	7	2.6.3	<code>\UntagName</code>	17
2.3.2	Forenames: <code>\FName</code>	8	2.6.4	<code>\ExcludeName</code> . .	18
2.4	Advanced Naming Topics	8	2.6.5	Indexing Certain Sections	18
2.4.1	Naming Conventions	8	2.6.6	Variant Spellings .	18
2.4.2	Hyphenation . . .	9	2.7	Naming Pattern Reference	19
2.4.3	Suffixes and Non-Western Names . .	9	2.7.1	Basic Naming . . .	19
2.4.4	Accented Names .	10	2.7.2	Suffixes	20
			2.7.3	Particles	20

1 Introduction

Book-length studies can engage more than 500 different names. Editors and proof-readers cost money when checking those names. This package provides basic tools for consistently and automatically formatting names for the body text and the index. The goal is to reduce error, streamline work, and become a more cost-effective author. Features include:

- Format, print, and index names all at once.
- Automatically print a full name the first time it appears, with full or shorter forms thereafter.
- Automatically apply typesetting to names, recalling German *Sperrdruck*.
- Rearrange text without retyping.
- Allow predictable name variants in the text, yet index consistent name forms.
- “Tag” and “untag” names in the index in order to index different people with the same name.
- Macros can be used in the body text, in a `\marginpar`, in tabular material, and in list environments like `enumerate`, among others.

1.1 Design Notes and Thanks

This package depends on `etoolbox`, `suffix`, and `xargs`. As of version 1.4 it has been refactored extensively with robustness and modularity in mind. It has been tested with `latex`, `lualatex`, `pdflatex` and `xelatex`. It will work with `makeindex` and `texindy`. This document was typeset with `pdflatex` and `makeindex`.

Thanks to MARC VAN DONGEN, ENRICO GREGORIO, PHILIPP STEPHANI, HEIKO OBERDIEK, UWE LUECK, and ROBERT SCHLICHT for their assistance. The light of their knowledge banished the umbra of my ignorance.

1.2 Disclaimer

This documentation uses names of historical figures because users refer to real people in their projects. At no time in this document am I intending either to promote, disparage, or make any assertions about any persons living or dead. All names mentioned herein deserve respect for the impact and legacy of their bearers.

1.3 Caveat

Throughout this manual I play a “dirty trick” that makes a name print as if it had not yet occurred. In some cases I make a first occurrence print as if the name already had occurred. This trick can be used, for example, to force the formatting of the first name in a chapter or section. See Section 2.4.7 for more.

2 Usage

2.1 Package Options

The defaults are `mainmatter`, `smallcaps`, `nocomma`, and `index`. The package begins immediately to format the first occurrences of names with small caps, suppressing commas between surnames and suffixes, and indexing all names in place. The following package options alter this behavior.

Enable/Disable Formatting

<code>mainmatter</code>	Enable formatting attributes (see below), starting at the beginning of a document.
<code>frontmatter</code>	Disable formatting <i>before</i> the invocation of <code>\NamesActive</code> while retaining automatic full and short forms. This option fits well, e.g., with a foreword from a contributor. See Section 2.4.6.

Show/Hide Suffix Commas

<code>nocomma</code>	Suppress commas between surnames and suffixes, following modern styles like <i>Chicago Manual of Style</i> . Commas can be forced on a per-use basis with <code>\ShowComma</code> . See Section 2.4.3.
<code>comma</code>	Retain commas between all surnames and suffixes. This imposes limits on the use of certain macros.

Enable/Disable Indexing

<code>index</code>	Create index entries in place with the names.
<code>noindex</code>	Prevent indexing before the invocation of the macro <code>\IndexActive</code> . See Section 2.6.5.

Formatting Attributes

<code>alwaysformat</code>	Disabled by default, this option can be used together with the options below. With the <code>mainmatter</code> option or after <code>\NamesActive</code> , always apply one of the predefined formatting options below or custom formatting (Section 2.4.5). This is quite similar to using <i>Sperrdruck</i> in German to call out names.
<code>smallcaps</code>	Set the first use of a name in small caps.
<code>italic</code>	Italicize the first occurrence of a name.
<code>boldface</code>	Set the first use of a name in boldface.
<code>noformat</code>	This formatting option creates no special formatting, but it does not alter the formatting mechanism like the <code>frontmatter</code> option.

2.2 Quick Start Guide

This general section gets one using about 60% of the package features right away. The seven basic patterns for names are:

Pattern	Restrictions
<code>\Name[(\Forenames)]{(\Surnames)}</code>	<i>none</i>
<code>\Name[(\Forenames)]{(\Surnames, Suffix)}</code>	<code>comma, nocomma</code>
<code>\Name[(\Forenames)]{(\Surnames)}[(\Nicknames)]</code>	<i>none</i>
<code>\Name[(\Forenames)]{(\Surnames, Suffix)}[(\Nicknames)]</code>	<code>comma, nocomma</code>
<code>\Name{(\Mononym)}</code>	<i>none</i>
<code>\Name{(\Mononym, Suffix or Sobriquet)}</code>	<code>comma, nocomma</code>
<code>\Name[(\Mononym)][(\Sobriquet)]</code>	<code>\AKA</code> and <code>\PName</code>

The `comma` and `nocomma` options affect the appearance of names. That is especially true for the last two forms above. These patterns apply to all macros that accept a name form, with some restrictions.

2.2.1 Hints for Easier Usage

- Q: There is too much space between the initials in the names.
- A: Bringhurst’s *Elements of Typographic Style* discourages full spaces between initials, calling for no spaces or thin spaces. Using `\frenchspacing` greatly simplifies related issues and it conforms with Bringhurst’s guidelines.
- Q: I see two index entries that are exactly the same!
- A: As of version 1.4, the effect of spurious spaces in the printed names is negligible, but those spaces cause duplicate index entries. If you follow those entries to their pages, chances are that you will be able to track down the offending errors in short order.
- Q: OK, you threw a bazillion macros at me, I used them all, and now I am Googling your personal info because I’m on a deadline and you will soon have a date with a Cyberman.
- A: Hop in your TARDIS and leave a sticky note to yourself in the past to start just with `\Name` and spend a few pages breaking it in. See what happens in the index too. Use the other macros only when you need a function that the current ones are not fulfilling.
- Q: `\Name[Davey]{Jones}[a Monkee]` shows “a Monkee Jones.”
- A: `\Name[(\Forenames)]{(\Surnames)}[(\Nickname)]` creates a nickname, not a sobriquet.
- Q: I see “Paragraph ended. . .” or “Missing *(grouping token)* inserted” and execution stops.
- A: Check if the `{braces}` and `[brackets]` are balanced.

- Q: `\Name[Henry]{VII}[Tudor]` prints either “Tudor VII” or “VII”; `\FName[Henry]{VII}[Tudor]` prints either “Tudor VII” or “Tudor.” May I hurt you now?
- A: The pattern `\Name[⟨King⟩]{⟨Suffix⟩}` fails because even though authorities like Mulvany, *Indexing Books* talk about optional surnames, it is far easier to encode royal names and mononyms as if they were surnames. Use either `\Name{Henry, VII}` with the `nocomma` option or `\Name{Henry}[VII]` with the `comma` option. Indicate that he was of House Tudor manually in the text and consider using something like `\TagName{Henry, VII}{ (Tudor)}` to indicate his house as part of the index entry.
- Q: `\Name{Aethelred, the really}[Unreadig]` shows “Aethelred the really Unreadig.”
- A: Its index entry is messed up too. The form `\Name{⟨Mononym, Suffix⟩}[⟨Sobriquet⟩]` creates two sobriquets, but badly. This is because the sobriquet feature of the final argument is best used with the `comma` option while the suffixed form is best used with the `nocomma` option. Mix them at your peril or at your whimsy.
- Q: I tried `\AKA{Boris}[the Animal]{Just Boris}` and that went about as well as getting my head used as a bowling ball.
- A: You cannot use `\AKA{⟨Mononym⟩}[⟨Sobriquet⟩]...` because that would call for a macro with an ambiguous argument list that forces adding extra explicit arguments to work properly. Then you have to remember if you did or did not — or should or should not — add the explicit macro, and by that time the point of diminishing returns would have passed for both the package author and the package users. Sorry, but the manual solution is better. That also applies to the pattern `\AKA[⟨Forenames⟩]{⟨Surnames⟩}[⟨Nicknames⟩]...` and the use of `\PName`.
- Q: So how do I deal with some stage names and the like?
- A: Use a forename or first initials to prevent failure:
 OK: `\Name[J.]{Kreskin}[The Amazing]`
 `\AKA[J.]{Kreskin}[Joseph]{Kresge}`
 FAIL: `\Name{Kreskin}[The Amazing]`
 `\AKA{Kreskin}[Joseph]{Kresge}`
- Q: `\Name` has a full name in the source, but it is shorter in the text!
- A: And you did not mean that to happen. Use `\Name*` in that case. It can be easy to forget that `\Name` can automatically shorten a reference even if you use the full name in the source text. Remember:
`\Name[John]{Smith}` gives JOHN SMITH
`\Name[John]{Smith}` gives Smith
`\FName[John]{Smith}` gives John

- Q: How do I refer to “Iron Mike” Tyson?
- A: Very, very carefully. Use ‘‘`\Name[Mike]{Tyson}[Iron Mike]`’’ and `\SubvertName[Mike]{Tyson}\Name[Mike]{Tyson}` to produce the example above. Or create two separate names and cross-reference them manually in order to track them both in the index (see page 15). Or use ‘‘`\AKA[Mike]{Tyson}{Iron Mike}`’’ if you just want to cross-reference the main name.
- Q: I had seen errors like “Argument of `\Name` has an extra `}`”
- A: Older versions of `nameauth` were susceptible to fragility and moving arguments. Version 1.4 and later of `nameauth` address many of these issues.

2.2.2 Error Handling

Except for the indexing parts of macros, most of the macros in this package print any erroneous arguments in the text, but not in the index. To find errors, one can look at index entries for irregularities or examine package warnings. Unless there is a syntax error, all macros emit meaningful warnings. Convenience macros like `\PName` produces warnings via their component macros `\Name` and `\AKA`.

Not all warnings are created equal. For example, the multiple creation of a cross-reference with `\AKA` will generate a warning, but it will have no ill effects and may be intentional. Other warnings, especially in the case of indexing macros, indicate that the macro produced no output.

Warnings result from:

1. Using a cross-reference `[\langle alternate FNN \rangle]{\langle alternate SNN \rangle}[\langle alt. names \rangle]` created by `\AKA` as a reference in `\Name`, `\FName`, and `\PName`.
2. Using a reference `[\langle FNN \rangle]{\langle SNN \rangle}[\langle alternate names \rangle]` created by `\Name`, `\FName`, and `\PName` as a cross-reference in `\AKA`.
3. Using `\AKA` to create the same cross-reference multiple times.
4. Using `\IndexName` to index a cross-reference as a main entry.
5. Using `\TagName` to tag a cross-reference.
6. Using `\ExcludeName` to exclude a name that has already been used.

2.3 Naming Macros

2.3.1 Surnames: `\Name` and `\Name*`

`\Name` This macro generates two forms of the name: a printed form in the text and a
`\Name*` form of the name that occurs in the index. The general syntax is:

```
\Name[⟨forename(s)⟩]{⟨surname(s)⟩}[⟨alternate names⟩]
\Name*[⟨forename(s)⟩]{⟨surname(s)⟩}[⟨alternate names⟩]
```

From now on we will abbreviate $\langle forename(s) \rangle$ with $\langle FNN \rangle$ and $\langle surname(s) \rangle$ with $\langle SNN \rangle$ at various points. The following table helps to show how the syntax description works with first and subsequent references:

<code>\Name[Albert]{Einstein}</code>	ALBERT EINSTEIN
<code>\Name*[Albert]{Einstein}</code>	Albert Einstein
<code>\Name[Albert]{Einstein}</code>	Einstein
<code>\Name{Confucius}</code>	CONFUCIUS
<code>\Name*{Confucius}</code>	Confucius
<code>\Name{Confucius}</code>	Confucius
<code>\Name[M.T.]{Cicero}[Marcus Tullius]</code>	MARCUS TULLIUS CICERO
<code>\Name*[M.T.]{Cicero}[Marcus Tullius]</code>	Marcus Tullius Cicero
<code>\Name[M.T.]{Cicero}[Marcus Tullius]</code>	Cicero
<code>\Name{Charles}[the Bald]</code>	CHARLES THE BALD
<code>\Name*{Charles}[the Bald]</code>	Charles the Bald
<code>\Name{Charles}[the Bald]</code>	Charles

`\Name` connects the $\langle FNN \rangle$ to the $\langle SNN \rangle$ to create respective printed and indexed forms, as illustrated in Section 2.7 and thereafter. The invocation of `\Name` always prints the $\langle SNN \rangle$ field. `\Name` prints the “full name” at the first occurrence, then only the partial form thereafter. `\Name*` always prints the full name.

Nicknames and sobriquets of some historical figures are implemented in slightly different ways. A nickname assumes the presence of $\langle FNN \rangle$ and $\langle SNN \rangle$ (see the example for Cicero above). The $\langle alternate names \rangle$ field allows a nickname to replace the $\langle FNN \rangle$ field in the text while keeping the $\langle FNN \rangle$ field in the index. Nicknames are truly optional. As long as the $\langle FNN \rangle$ and $\langle SNN \rangle$ fields are consistent, the index entries will be consistent. The shorter name printed by `\Name` will not print forenames or nicknames. See also Section 2.2 and `\FName` below.

In the example of Charles the Bald above, “the Bald” is not a nickname because no $\langle FNN \rangle$ are present. In that case the $\langle alternate names \rangle$ field is appended to the $\langle SNN \rangle$ in both the printed form *and* in the index form. This behavior implies that, while “the Bald” uses an optional parameter, it is not truly optional once used.

Later we shall see that a similar effect, perhaps more preferable, can be achieved with the default `nocomma` option and `\Name{Charles, the Bald}`, as seen in Section 2.4.3. One must always be consistent in using these forms. Otherwise one will get bogus index entries and errors in the text that are difficult to track.

2.3.2 Forenames: \FName

\FName This casual friend of **\Name** prints only “first” names, but it will still print a full name when a first use occurs. **\FName** is intended for Western-style names. **\FName*** is only a synonym for **\FName**. The syntax is basically the same:

`\FName[⟨FNN⟩]{⟨SNN⟩}[⟨alternate names⟩]`

The following table shows the output at a glance:

<code>\FName[Albert]{Einstein}</code>	ALBERT EINSTEIN
<code>\FName[Albert]{Einstein}</code>	Albert
<code>\FName{Confucius}</code>	CONFUCIUS
<code>\FName{Confucius}</code>	Confucius
<code>\FName[M.T.]{Cicero}[Marcus Tullius]</code>	MARCUS TULLIUS CICERO
<code>\FName[M.T.]{Cicero}[Marcus Tullius]</code>	Marcus Tullius
<code>\FName{Charles}[the Bald]</code>	CHARLES THE BALD
<code>\FName{Charles}[the Bald]</code>	Charles

See how the first reference is a full name? That prevents an accidental reference to a first name before a person has been introduced. Nicknames are used by including them in the *⟨alternate names⟩* field in addition to *⟨FNN⟩*. For example, aviation hero CHESLEY B. SULLENBERGER III can be noted as:

`‘‘\FName[Chesley B.]{Sullenberger, III}[Sully]’’` “Sully”

A good way to cut keystrokes would be to assign the above macro to the control sequence **\Sully**. With comma-delimited suffixes we note special cases governed by the **nocomma** package option (see Section 2.4.3). These include names like the fictional J.D. ROCK III and the real CHARLES V:

`\FName[J.D.]{Rock, III}, “J.D.”; \FName{Charles, V}, “Charles.”`

\FName suppresses extra periods if a forename with initials occurs at the end of a sentence, as in `‘‘who shot \FName[J.D.]{Rock, III}.’’` “who shot J.D.”

2.4 Advanced Naming Topics

2.4.1 Naming Conventions

\CapThis English names with the particles *de*, *de la*, *d’*, *von*, *van*, and *ten* generally keep them with the last name, using varied capitalization. *Le*, *La*, and *L’* are capitalized unless preceded by *de*. In English, these particles go in the *⟨SNN⟩* field of **\Name**, e.g., WALTER DE LA MARE. To capitalize the first particle in a subsequent **\Name** reference at the beginning of a sentence, use **\CapThis\Name[Walter]{de la Mare}**. De la Mare will think it fair. DU CANGE (Charles du Fresne) would too, because **\CapThis** now works for all instances of **\Name** as of version 1.4.

Names foreign to English usually put these particles in the *⟨FNN⟩* field of **\Name**. Yet these particles are not first names. Using **\FName** with alternate forenames avoids that issue. See the examples starting on page 21.

2.4.2 Hyphenation

I find it helpful to use respectively the `babel` or `polyglossia` packages with name hyphenation. If one is using English as the main language, the default hyphenation patterns may not suffice. For example, the name JOHN STRIETELMEIER may break thus: “Stri-etelmeier.” That is fixed by creating a `\de` macro equivalent to `\newcommand{\de}[1]{\foreignlanguage{ngerman}{#1}}` (using `babel`) and writing `\de{\Name[John]{Strietelmeier}}`.

One can insert optional hyphens in the arguments of `\Name` and friends but that must be done *consistently* to avoid variants being treated as different names.

2.4.3 Suffixes and Non-Western Names

`\ShowComma` This package has two main settings: `comma` and `nocomma`. These settings are meant to accommodate different typesetting conventions. When using the default `nocomma` option, the user can force a comma to appear between the name and suffix in the $\langle SNN \rangle$ field as appropriate by putting `\ShowComma` before `\Name` and friends, such as `\ShowComma\Name[Louis]{Gossett, Jr.}` LOUIS GOSSETT, JR.

The `comma` option restricts the flexibility of expressing some royal names and basic Eastern names. `\AKA` and `\PName` cannot create cross-references to these forms, but `\AKA` can cross-reference *from* these forms in its second set of arguments. These restricted forms are shown below:

<code>\Name{Henry}[VIII]</code>	HENRY VIII
<code>\Name{Henry}[VIII]</code>	Henry
<code>\Name{Chiang}[Kai-shek]</code>	CHIANG KAI-SHEK
<code>\Name{Chiang}[Kai-shek]</code>	Chiang

Again, alternate forenames *override* the $\langle FNN \rangle$ in the text. Sobriquets are *appended* to $\langle SNN \rangle$. The presence or absence of $\langle FNN \rangle$ triggers this difference between the two actions— **this is a central concept**.

There is perhaps a less-confusing manner of working with such names. `\Name`—not `\Name*`—truncates comma-delimited suffixes from last names. For example, it prints the name OSKAR HAMMERSTEIN II the first time and Hammerstein thereafter. One must always use a comma to activate this, e.g., `\Name[Oskar]{Hammerstein, II}`. Spaces (or no spaces) between the comma and the suffix must remain consistent or else they will produce duplicate index entries. Extra space does not, however, affect the formatted output.

Again, **the comma is essential to how this feature works**. Using, e.g., `\Name{Base, suff1, suff2}` for the $\langle SNN \rangle$ argument yields: BASE SUFF1, SUFF2; then Base. See how only the first comma is suppressed.

These macros keep track of whether the name ends with the period of an abbreviation like “Jr.” and “Sr.” That should also work with abbreviations like “d. Ä.” (*der Ältere*). Two periods are not printed when the full name is printed at the end of a sentence. The following example shows the combinations:

<code>\Name[Martin Luther]{King, Jr.}</code>	MARTIN LUTHER KING JR.
<code>\Name[Martin Luther]{King, Jr.}</code>	King.
<code>\Name[Martin Luther]{King, Jr.}</code>	King (e.g., in a sentence)
<code>\Name*[Martin Luther]{King, Jr.}</code>	Martin Luther King Jr.
<code>\Name*[Martin Luther]{King, Jr.}</code>	Martin Luther King Jr.

Using the default package option `nocomma` with suffix removal, one can use the suffix feature to tweak more possibilities out of `\Name`. Instead of the sobriquet feature, one could use the following variants:

<code>\Name{Louis, XIV}</code>	LOUIS XIV
<code>\Name{Louis, XIV}</code>	Louis
<code>\Name{Sun, Yat-sen}</code>	SUN YAT-SEN
<code>\Name{Sun, Yat-sen}</code>	Sun

The benefit to using this form is that one can type `\Name*{Louis, XIV}`, the `“\AKA{Louis, XIV}{Sun King}”` and get Louis XIV, the “Sun King” in the text with an appropriate reference from “Sun King” to “Louis XIV” in the index. The sobriquet feature would otherwise prevent such usage.

Even though suffix and sobriquet features look like they produce the same *output* in the body text using the `nocomma` option, they are internally *different*. They will not respect each other regarding “first use,” although they will (tentatively) cooperate in the index. Use each approach consistently. An example of “dangerous” use of these features occurs in Section 2.4.5.

2.4.4 Accented Names

If you use accented names, `xindy` (or `texindy`) are strongly recommended. The following Unicode characters are available using `inputenc`/`fontenc`:

À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	FIRST USE
À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	second use
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ ß	FIRST USE
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ ß	second use
À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	FIRST USE
à á â ã ä å æ	ç è é ê ë	ì í î ï ð ñ	second use
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ ÿ	FIRST USE
ò ó ô õ ö ø	ù ú û ü ý	þ ÿ	second use
Ă Ǻ Ȧ Ȧ Ć ć Č č	Ď ě Đ đ Ě ě Ě ě	Ǧ ǧ Ĩ ĩ	FIRST USE
Ă Ǻ Ȧ Ȧ Ć ć Č č	Ď đ Đ đ Ě ě Ě ě	Ǧ ǧ Ĩ ĩ	second use
Ĳ Ĳ Ĳ Ĳ Ĳ Ĳ	Ń ń Ń ń Œ œ	Ř ř Ř ř	FIRST USE
Ĳ Ĳ Ĳ Ĳ Ĳ Ĳ	Ń ń Ń ń Œ œ	Ř ř Ř ř	second use
Š š Š š Ť ť Ť ť	Ů ů Ů ů	Ž ž Ž ž Ž ž	FIRST USE
Š š Š š Ť ť Ť ť	Ů ů Ů ů	Ž ž Ž ž Ž ž	second use

With `makeindex` there are a number of solutions, but some depend on the availability of the `-g` option and user-created settings in an `.ist` file whose description goes beyond the scope of this document. You also could use `\IndexActive` and `\IndexInactive` to suppress indexing, then create manual entries. For non-English documents, however, that solution is not optimal.

One may use control sequences in `\Name` (thanks Robert Schlicht). That means more accents with NFSS, such as the next example that uses `inputenc`/`fontenc`:

```
\usepackage{newunicodechar}
\DeclareTextSymbolDefault{\textlong{s}}{TS1}
\DeclareTextSymbol{\textlong{s}}{TS1}{115}
\newunicodechar{f}{\textlong{s}}
\newunicodechar{ā}{\=a}
\newunicodechar{ṁ}{\d m}
```

For some situations, such as “traditional” NFSS, you will need fonts with TS1 glyphs, e.g., `\usepackage{lmodern}`. See the informative tables on pages 455–63 in *The LaTeX Companion*. This allows `\Name{Ghazāli}` to generate GHAZĀLĪ.

In some cases, indexing accented names will fail unless you use `xetex` or `luatex`, both of which use `fontspec` instead of NFSS. Control sequences like `\=a` fail even in manual index entries when using `makeindex` and `gind.ist` because the equal sign is interpreted as a “literal” character, as mentioned by DAN LUECKING. I used `\IndexInactive\Name{Ghazāli}\IndexActive` to prevent the index entry “ali” sorted under “Ghaz” because even the manual entry fails in that case.

It is important that this package work in the context of multiple languages. The use of multiple typesetting engines facilitates that. This snippet from the preamble to this file allows it to be typeset with multiple engines.

```
\usepackage{ifxetex}
\usepackage{ifluatex}
\ifxetex % uses fontspec
  \usepackage{fontspec}
  \defaultfontfeatures{Mapping=tex-text}
  \usepackage{xunicode}
  \usepackage{xltextra}
\else
  \ifluatex % also uses fontspec
    \usepackage{fontspec}
    \defaultfontfeatures{Ligatures=TeX}
  \else % traditional NFSS
    \usepackage[utf8]{inputenc}
    \usepackage[TS1,T1]{fontenc}
  \fi
\fi
```

I used the following in the text itself to allow for conditional processing that helps one to document work under multiple engines:

```
\ifxetex <xelatex text>%
\else%
  \ifluatex%
    \ifpdf <lualatex in pdf mode text>%
    \else <lualatex in dvi mode text>%
    \fi%
  \else%
    \ifpdf <pdf latex text>%
    \else <latex text>%
    \fi%
  \fi%
\fi
```

2.4.5 Name Formatting

`\NamesFormat` The first instance of a name is formatted with `\NamesFormat` when formatting is active. Additionally, the `alwaysformat` option will cause every name to be formatted when formatting is active. In addition to using the package options, one can redefine `\NamesFormat` to create some custom effects. For example, if you wanted to suppress formatting in footnotes, you could do something like:

```
\makeatletter
\let\@oldfntext\@makefntext
\long\def\@makefntext#1{\def\NamesFormat{}\@oldfntext{#1}}
\makeatother
```

This approach synchronizes the “first use” feature in the text and the footnotes, but only suppresses the formatting. It takes advantage of the deep nesting of `\@makefntext` and a localized `\def` to make a temporary change. A second example puts the mention of first names in the margin if possible:

```
\let\oldformat\NamesFormat
\renewcommand{\NamesFormat}[1]{\textbf{#1}%
\ifinner\else\marginpar{\scriptsize #1}\fi}
```

The result produces something like:

```
\Name{Vlad III}[Dracula] was known as Vlad Țepeș, "The Impaler,"
after his death. He was the son of \Name{Vlad II}% [Dracul], a
member of the Order of the Dragon. Later references to
\Name{Vlad III}[Dracula] appear thus.
```

Vlad III Dracula was known as Vlad Țepeș, “The Impaler,” after his death. He was the son of **Vlad II Dracul**, a member of the Order of the Dragon. Later references to Vlad III appear thus.

Vlad III Dracula
Vlad II Dracul

The forms above do not work with `\PName` and `\AKA`. Consistently use either the suffix mechanism (Section 2.4.3) or see Section 2.5.1 regarding manual entries. If you use the suffix mechanism, you would use the following forms:

```
\Name{Vlad III, Dracula} was known as \AKA{Vlad III, Dracula}%
{Vlad}[Țepeș], ‘‘\AKA*{Vlad III, Dracula}{Vlad}[the Impaler],’’
after his death. He was the son of \Name{Vlad II, Dracul}, a
member of the Order of the Dragon. Later references to
\Name{Vlad III, Dracula} appear thus.
```

Vlad III Dracula was known as Vlad Țepeș, “the Impaler,” after his death. He was the son of **Vlad II Dracul**, a member of the Order of the Dragon. Later references to Vlad III appear thus.

The redefinition of `\NamesFormat` given above follows scoping rules. Since I used it in a quote environment, it reverts to normal and now we have: VLAD III DRACULA. Later references produce Vlad III. The “Dracula” example was manipulated extensively with the tweaking macros `\ForgetName` and `\SubvertName`. Mixing the sobriquet and suffix forms otherwise might cause errors to “bite.”

2.4.6 Formatting Certain Sections

`\NamesActive` Using the `frontmatter` option deactivates formatting until `\NamesActive` occurs.
`\NamesInactive` Another macro, `\NamesInactive`, will deactivate formatting again. These two macros toggle two independent systems of formatting and first use.

Here we switch to the “front matter” mode with `\NamesInactive`:

```
\Name[Rudolph]{Carnap}      Rudolph Carnap
\Name[Rudolph]{Carnap}      Carnap
\Name[Nicolas]{Malebranche} Nicolas Malebranche
\Name[Nicolas]{Malebranche} Malebranche
```

Then we switch back to “main matter” mode with `\NamesActive`:

```
\Name[Rudolph]{Carnap}      RUDOLPH CARNAP
\Name[Rudolph]{Carnap}      Carnap
\Name[Nicolas]{Malebranche} NICOLAS MALEBRANCHE
\Name[Nicolas]{Malebranche} Malebranche
```

Notice that we have two independent cases of “first use” above. That is intended for different sections of the document, like front matter and main matter. It clashes on the same page. That is why you should suppress formatting in the main matter with a custom definition of `\NamesFormat` (Section 2.4.5) instead of using `\NamesInactive`. The following section covers tweaks that help to minimize certain first-use clashes.

2.4.7 Tweaks: `\ForgetName` and `\SubvertName`

These two macros are meant for tweaking text at or near final draft stage. They affect both front matter and main matter.

`\ForgetName` This macro is a “dirty trick” of sorts that takes the same optional and mandatory parameters used by `\Name`. It handles its arguments in the same way, except that it ignores the final parameter if `\FNN` are present. The syntax is:

`\ForgetName[\FNN]{\SNN}[\alternate names]`

This macro causes `\Name` and friends to “forget” prior uses of a name with respect to typesetting. The next use will print as if it were a “first use.” Index entries and cross-references (see the next section) are *never* forgotten.

`\SubvertName` This macro is the opposite of the one above. It takes the same parameters. It handles its arguments in the same manner. The syntax is:

`\SubvertName[\FNN]{\SNN}[\alternate names]`

This macro causes `\Name` and friends to think that a prior use of a name already has occurred. The next use will print as if it were a “subsequent use.”

2.5 Name Variant Macros

2.5.1 Introduction

`\AKA` handles pseudonyms, stage names, *noms de plume*, and so on. Before we examine `\AKA` in detail, we should cover the manual solution that `\AKA` helps to automate. If needed, the `.idx` file can be a helpful reference when linking manual entries with `nameauth` entries. This remains the only solution for certain cases, especially with the `comma` option:

```
\index{Jean the Fearless|see{Jean sans Peur}}%
\Name{Jean}[sans Peur] (Jean the Fearless) reigned as Duke of
Burgundy from 1404 to 1419.
JEAN SANS PEUR (Jean the Fearless) reigned as Duke of Burgundy
from 1404 to 1419.
```

The suffix workaround would use `\Name{Jean, sans Peur}` and the author need merely type `\AKA{Jean, sans Peur}{Jean the Fearless}`. See also Section 2.4.3. Using `makeindex` also requires some manual entries:

```
\index{Doctor Angelicus@\textit{Doctor Angelicus}|see{Thomas Aquinas}}
\index{Thomas of Aquino|see{Thomas Aquinas}}
Perhaps the greatest medieval theologian was \Name{Thomas}[Aquinas]
(Thomas of Aquino), also known as \textit{Doctor Angelicus}. His name
"Aquinas" is not a surname.
Perhaps the greatest medieval theologian was THOMAS AQUINAS
(Thomas of Aquino), also known as Doctor Angelicus. His name
"Aquinas" is not a surname.
```

2.5.2 \AKA

\AKA The primary macro that handles aliases is \AKA. Its syntax is:
\AKA*

\AKA[⟨FNN⟩]{⟨SNN⟩}[⟨alt. FNN⟩]{⟨alt. SNN⟩}[⟨alt. names⟩]
\AKA*[⟨FNN⟩]{⟨SNN⟩}[⟨alt. FNN⟩]{⟨alt. SNN⟩}[⟨alt. names⟩]

The ⟨FNN⟩ and ⟨SNN⟩ arguments *do not include* the final optional argument of \Name to avoid ambiguity. \AKA will not create a *see* reference to a name that uses the ⟨alternate names⟩ field. That must be done manually (see above). This macro is designed mainly with Western names in mind.

\AKA only prints the alternate name, not the name to which it refers. It assumes that a \Name macro occurs somewhere to create the page-indexed target of a cross-reference. **No error checking** occurs for this. The macro also prevents double periods. Following is a quick review of what works and what fails for examples using BOB HOPE, LOUIS XIV, and GREGORY I:

\AKA[Bob]{Hope}[Leslie Townes]{Hope}	Leslie Townes Hope
†\AKA[Bob]{Hope}[Leslie Townes]{Hope}[Lester T.]	Lester T. Hope
‡\AKA{Louis, XIV}{Sun King}	Sun King
\AKA{Louis}[XIV]{Sun King}	FAIL
§\AKA{Gregory, I}{Gregory}[the Great]	Gregory the Great
\AKA*{Gregory, I}{Gregory}[the Great]	the Great
\AKA{Gregory}[I]{Gregory}[the Great]	FAIL

† This succeeds, but replaces “Leslie Townes” with “Lester T.” in the text, while keeping “Hope, Leslie Townes” as the *see* reference in the index.

‡ This form uses the `nocomma` feature.

§ This produces different output, depending on whether \AKA or \AKA* is used.

\AKA prints an alternate name and creates a cross reference in the index. The target of this cross-reference is either ⟨SNN⟩, ⟨FNN⟩ or just ⟨SNN⟩. In the cross-reference, the ⟨alternate names⟩ replace the ⟨alternate FNN⟩ if both exist. Otherwise the ⟨alternate names⟩ follow the ⟨alternate SNN⟩:

Today we consider \AKA[George]{Eliot}[Mary Anne]{Evans} and
her literary contributions as \Name[George]{Eliot}.

Today we consider Mary Anne Evans and her literary contributions as
GEORGE ELIOT.

If the starred form \AKA* is used with the template ⟨none⟩⟨alternate SNN⟩⟨alt. names⟩, it only prints the ⟨alt. names⟩. With the same template, \AKA prints ⟨alternate SNN⟩ followed by ⟨alt. names⟩. Section 2.6.2 offers a complete presentation of related features.

The cross-references generated by \AKA and \AKA* are meant to be *see* references and thus the other macros in this package will not create page references for these cross-references. See also Section 2.2.2. In certain cases, the alternate name might need to be indexed with page numbers. Do not use \AKA in those cases. Use \Name for both the main and the alternate names. Then, only after both instances of \Name have been invoked, create manual cross-references with \index, e.g.:

Authoritative Name	Alternate Name	Example of Use
MAIMONIDES	Moses ben-Maimon	<code>\Name{Maimonides}</code> <code>\AKA{Maimonides}{Moses ben-Maimon}</code>
Maimonides	RAMBAM	<code>\Name{Rambam}%</code> <code>\index{Rambam seealso{Maimonides}}</code>

`\AKA` will not create multiple instances of a cross-reference. This allows the macro `\ExcludeName` to work, but it also prevents the special case where one moniker applies to multiple people, e.g.: WILLEBRORD SNEL VAN ROYEN (Snellius) and his son RUDOLPH SNEL VAN ROYEN (Snellius). `\AKA` produces the first cross-reference; the user manually creates the second:

```
\index{Snellius|see{Snel van Royen, Rudolph}}
```

2.5.3 `\PName`

`\PName` `\PName` is a “convenience macro” that sacrifices flexibility for simplicity. It uses only the $\langle FNN \rangle$ $\langle SNN \rangle$ use of `\AKA`. It does not use `\AKA*`. It is meant for Western-style names. It calls formats a Western-style “main” name followed by a cross-reference in parentheses. The syntax is:

```
\PName[ $\langle FNN \rangle$ ]{ $\langle SNN \rangle$ }[ $\langle other FNN \rangle$ ]{ $\langle other SNN \rangle$ }[ $\langle other alt. \rangle$ ]
```

The author determines the name that is indexed (the first name) and the subsequent name that only occurs as a *see* reference. For example:

<code>\PName[Mark]{Twain}[Samuel L.]{Clemens}</code>	MARK TWAIN (Samuel L. Clemens)
<code>\PName*[Mark]{Twain}[Samuel L.]{Clemens}</code>	Mark Twain (Samuel L. Clemens)
<code>\PName[Mark]{Twain}[Samuel L.]{Clemens}</code>	Twain (Samuel L. Clemens)
<code>\PName{Voltaire}[François-Marie]{Arouet}</code>	VOLTAIRE (François-Marie Arouet)
<code>\PName*{Voltaire}[François-Marie]{Arouet}</code>	Voltaire (François-Marie Arouet)
<code>\PName{Voltaire}[François-Marie]{Arouet}</code>	Voltaire (François-Marie Arouet)

2.6 Indexing Macros

2.6.1 `\IndexName`

`\IndexName` This macro creates an index entry like those created by `\Name` and friends. It prints no text in the body and permits no special formatting. The syntax is:

```
\IndexName[ $\langle FNN \rangle$ ]{ $\langle SNN \rangle$ }[ $\langle alternate names \rangle$ ]
```

`\IndexName` does not index $\langle alternate names \rangle$ unless $\langle FNN \rangle$ are absent, whereupon it treats $\langle alternate names \rangle$ as a sobriquet. If indexing is switched off (see Section 2.6.5), this macro does nothing. It will not create index entries for names that have been used as cross-references.

2.6.2 \TagName

\TagName This macro creates a tag that will appear in all index entries corresponding to the name arguments from the point of invocation of **\TagName** onward. For consistency throughout the document, tag names at the beginning. The syntax is:

`\TagName[<FNN>]{<SNN>}[<alternate names>]{<tag>}`

\TagName and **\UntagName** handle their name arguments like **\IndexName**. Tags created by **\TagName** can be helpful in the indexes of history texts. Several features of this package are designed for historical research. Suppose you are working with medieval subject matter. The following macros come in handy:

<code>\TagName{Leo, I}{, pope}</code>	Tag these names at the beginning
<code>\TagName{Gregory, I}{, pope}</code>	of the document.
...	
<code>\Name{Leo, I}</code>	First references to LEO I and
<code>\Name{Gregory, I}</code>	GREGORY I
...	
<code>\Name*{Leo, I} was known as</code>	Leo I was known as Leo the
<code>\AKA{Leo, I}{Leo}[the Great].</code>	Great.
...	
<code>\Name{Gregory, I}</code>	Gregory “the Great” was another
<code>‘\AKA*{Gregory, I}%</code>	major pope.
<code>{Gregory}[the Great]’</code>	

\TagName causes tags in the index entries to be inserted automatically from the point of invocation. **\AKA** prints the name and the sobriquet, while **\AKA*** only prints the sobriquet. The tag is literal text; it can be comma-delimited, in parentheses, or whatever an author might want.

2.6.3 \UntagName

\UntagName This macro removes a tag created by **\TagName**. The syntax is:

`\UntagName[<FNN>]{<SNN>}[<alternate names>]`

\TagName will replace one tag with another tag, but it does not remove a tag from a name. That is the role of **\UntagName**. By using these two commands, one can disambiguate different people with the same name. For example:

<code>\ForgetName[John]{Smith}</code>	This is the first reference to JOHN
<code>\TagName[John]{Smith}%</code>	SMITH, “the other one” in the in-
<code>{, the other one}</code>	dex. Notice that it is tweaked.
<code>\Name[John]{Smith}...</code>	
<code>\ForgetName[John]{Smith}</code>	This refers to JOHN SMITH “the
<code>\TagName[John]{Smith}%</code>	third” in the index. It is tweaked
<code>{, the third}</code>	again as a “first occurrence.”
<code>\Name[John]{Smith}...</code>	
<code>\SubvertName[John]{Smith}</code>	This refers to the John Smith in
<code>\UntagName[John]{Smith}</code>	Section 2.2. It is tweaked in order
<code>\Name*[John]{Smith}</code>	to force a subsequent reference.

Tagging and untagging in this manner requires the author to juggle more info. With more freedom comes more responsibility.

2.6.4 `\ExcludeName`

`\ExcludeName` This prevents `\Name`, etc. from both formatting and indexing a specific name, but *only if that name has not been used*. See also Section 2.2.2. The syntax is:

`\ExcludeName[$\langle FNN \rangle$]{ $\langle SNN \rangle$ }[alternate names]`

To suppress only indexing but retain formatting, enclose `\Name`, etc. between `\IndexInactive` and `\IndexActive`.

2.6.5 Indexing Certain Sections

`\IndexActive` Using the `noindex` option deactivates indexing until `\IndexActive` occurs. An-
`\IndexInactive` other macro, `\IndexInactive`, will deactivate indexing again. These can be used throughout the document, independently of `\ExcludeName`.

2.6.6 Variant Spellings

Handling variant name spellings can be complicated, but one could create macros on a per-case basis to make it easier. For example, one might settle on the form W.E.B. DU BOIS in one’s name authority. Yet an essay could use W.E.B. DuBois, where the publisher would not grant the right to alter the spelling. In that case, do the following in that document section:

1. In those cases where the only variation in the name is spacing (as above), you must call `\ForgetName` to generate a “first use” of the alternate spelling. The “first use” mechanism ignores spaces.
2. In all cases of the variant spelling, wrap `\Name` and friends between `\IndexInactive` and `\IndexActive`. A macro can do this easily.
3. Call `\IndexName` with the authoritative form right after `\IndexActive`. Again, this can be part of a macro.

This looks cumbersome, but it ensures accuracy. It cannot be reduced to an all-purpose macro because that would generate an ambiguous argument list. It should only be used in those cases where minor variations in spelling do not cause the reader to question the identity of the person in question.

2.7 Naming Pattern Reference

2.7.1 Basic Naming

The macro used most often is `\Name`, but we list the starred form first here because it always produces the fullest name form. When referring to a name for the first time, the following macros have the same result:

1. First reference: JOHN SMITH

```
\Name*[John]{Smith} or
\Name[John]{Smith} or
\FName[John]{Smith}
```

2. First mononym reference: PLATO

```
\Name*{Plato} or
\Name{Plato} or
\FName{Plato}
```

Subsequent references to names have a different format depending on the macro used:

3. Subsequent full name: John Smith

```
\Name*[John]{Smith}
```

4. Subsequent surname: Smith

```
\Name[John]{Smith}
```

5. Subsequent forename: John

```
\FName[John]{Smith}
```

6. Subsequent mononym: Plato

```
\Name*{Plato} or
\Name{Plato} or
\FName{Plato}
```

These examples show printing of alternate forenames, keeping the same name in the index. They use both the first and final optional parameters of `\Name` and friends.

7. Long first ref: JANE Q. PUBLIC

```
\Name*[J.Q.]{Public}[Jane Q.] or
\Name[J.Q.]{Public}[Jane Q.] or
\FName[J.Q.]{Public}[Jane Q.]
```

8. Different forenames, same surname: Jane Qetsiyah Public

```
\Name*[J.Q.]{Public}[Jane Qetsiyah]
```

9. Subsequent name: J.Q. Public

```
\Name*[J.Q.]{Public}
```

10. Alternate forename: Janie

```
\FName[J.Q.]{Public}[Janie]
```

These next examples work with or without the comma option. `\AKA` and `\PName` cannot cross-reference these forms. Page 20, Section 2.4.3, and Section 2.5.1 address this issue with different solutions.

11. First Eastern ref: MAO TSE-TUNG

```
\Name*{Mao}[Tse-tung] or
\Name{Mao}[Tse-tung]
```

12. Subsequent references: Mao Tse-tung

```
\Name*{Mao}[Tse-tung]
```

13. Subsequent references: Mao

```
\Name{Mao}[Tse-tung] or
\FName{Mao}[Tse-tung]
```

14. First royal: LOUIS THE PIOUS

```
\Name*{Louis}[the Pious] or
\Name{Louis}[the Pious] or
\FName{Louis}[the Pious]
```

15. Subsequent refs: Louis the Pious

```
\Name*{Louis}[the Pious]
```

16. Subsequent references: Louis

```
\Name{Louis}[the Pious] or
\FName{Louis}[the Pious]
```

17. First ancient: PTOLEMY I SOTER

```
\Name*{Ptolemy I}[Soter] or
\Name{Ptolemy I}[Soter] or
\FName{Ptolemy I}[Soter]
```

- | | |
|--|--|
| 18. Subsequent refs: Ptolemy I Soter
<code>\Name*{Ptolemy I}{Soter}</code> | <code>\Name{Henry}[VIII]</code> or
<code>\FName{Henry}[VIII]</code> |
| 19. Subsequent refs: Ptolemy I
<code>\Name{Ptolemy I}{Soter}</code> or
<code>\FName{Ptolemy I}{Soter}</code> | 21. Subsequent refs: Henry VIII
<code>\Name*{Henry}[VIII]</code> |
| 20. First royal: HENRY VIII
<code>\Name*{Henry}[VIII]</code> or | 22. Subsequent references: Henry
<code>\Name{Henry}[VIII]</code> or
<code>\FName{Henry}[VIII]</code> |

2.7.2 Suffixes

Always use a comma to delimit suffixes. `\AKA` and `\PName` will cross-reference these forms. See also Section 2.4.3. Macros that are too wide for the column are split using the comment token for readability.

1. First: GEORGE S. PATTON JR.
`\Name*[George S.]{Patton, Jr.}` or
`\Name[George S.]{Patton, Jr.}` or
`\FName[George S.]{Patton, Jr.}`
2. Subsequent full: George S. Patton Jr.
`\Name*[George S.]{Patton, Jr.}`
3. Subsequent surname: Patton
`\Name[George S.]{Patton, Jr.}`
4. Subsequent forename: George
`\FName[George S.]{Patton, Jr.}%`
`[George]`

The next cases depend on the default `nocomma` option, whereby one can use comma suppression to implement forms of ancient, royal and Eastern names. `\AKA` and `\PName` will cross-reference these forms.

Cf. the reference to Ptolemy I (above). Using `\Name{Demetrius, I Soter}` keeps the number with the suffix. To keep the number with the name, use `\Name{Demetrius I, Soter}`.

2.7.3 Particles

The following illustrate the American style of particulate names.

1. First: WALTER DE LA MARE
`\Name*[Walter]{de la Mare}` or

5. First reference: FRANCIS I
`\Name*[Francis, I]` or
`\Name{Francis, I}` or
`\FName{Francis, I}`
6. Subsequent full name: Francis I
`\Name*[Francis, I]`
7. Subsequent name: Francis
`\Name{Francis, I}` or
`\FName{Francis, I}`
8. First reference: DEMETRIUS I SOTER
`\Name*[Demetrius, I Soter]` or
`\Name{Demetrius, I Soter}` or
`\FName{Demetrius, I Soter}`
9. Next full name: Demetrius I Soter
`\Name*[Demetrius, I Soter]`
10. Subsequent name: Demetrius
`\Name{Demetrius, I Soter}` or
`\FName{Demetrius, I Soter}`
11. First reference: SUN YAT-SEN
`\Name*[Sun, Yat-sen]` or
`\Name{Sun, Yat-sen}` or
`\FName{Sun, Yat-sen}`
12. Subsequent full name: Sun Yat-sen
`\Name*[Sun, Yat-sen]`
13. Subsequent name: Sun
`\Name{Sun, Yat-sen}` or
`\FName{Sun, Yat-sen}`

`\Name[Walter]{de la Mare}` or
`\FName[Walter]{de la Mare}`

2. Next reference: de la Mare
`\Name[Walter]{de la Mare}`

3. At start of sentence: De la Mare
`\CapThis\Name[Walter]{de la Mare}`
4. Forename: Walter
`\FName[Walter]{de la Mare}`

The Continental style differs slightly. Forms 5–7 put the particles in the index. Forms 8–11 use the nickname feature to omit the particles from the index. Long macros are split for readability.

5. The (admittedly long) first use:
 JOHANN WOLFGANG VON GOETHE
`\Name*[Johann Wolfgang von]%
 {Goethe}` or
`\Name[Johann Wolfgang von]%
 {Goethe}` or
`\FName[Johann Wolfgang von]%
 {Goethe}`

6. Subsequent: Goethe
`\Name[Johann Wolfgang von]%
 {Goethe}`
7. Forenames: Johann Wolfgang
`\FName[Johann Wolfgang von]%
 {Goethe}[Johann Wolfgang]`
8. First: ADOLF VON HARNACK
`\Name*[Adolf]{Harnack}[Adolf von]` or
`\Name[Adolf]{Harnack}[Adolf von]` or
`\FName[Adolf]{Harnack}[Adolf von]`
9. Next full name: Adolf von Harnack
`\Name*[Adolf]{Harnack}[Adolf von]`
10. Subsequent surname: Harnack
`\Name[Adolf]{Harnack}[Adolf von]`
 or `\Name[Adolf]{Harnack}`
11. Subsequent forename: Adolf
`\FName[Adolf]{Harnack}`

Change History

v0.7	General: Initial version	1	v0.96	General: Bugfixes	1
v0.75	General: New features added and described	1	v1.0	General: Works fully with <code>microtype</code> and <code>memoir</code>	1
v0.8	General: Improved function and compatibility; added quick start guide	1	v1.1	General: Fixed errors when emitting warnings	1
v0.85	General: Added comma suppression, new class options, and more functionality	1	v1.2	General: Added tagging features; extensively edited documentation	1
v0.86	General: Fixed some regressions . .	1	v1.22	General: Attempted first-use particle caps	1
v0.9	General: Added first name formatting; comma and suffix handling expandable	1	v1.24	General: Reverted functionality and suggested <code>\CapThis</code> workaround	1
v0.92	General: Build with all major <code>L^AT_EX</code> engines	1	v1.26	General: Fixed sorting of name suffixes in index	1
v0.94	General: Added index suppression, error checking, name particle caps	1	v1.4	General: Fixed issues moving argument problems, added several features	1
v0.95	General: Bugfixes	1			

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

A		G	Malebranche, Nicolas 13
\AKA	15	Goethe, Johann Wolf-	Mao Tse-tung 19
\AKA*	15	gang von 21	Moses ben-Maimon . .
Arouet, François-Marie		Gossett, Louis, Jr. 9 <i>see</i> Maimonides
. <i>see</i> Voltaire		Gregorio, Enrico 2	
C		Gregory I, pope 15, 17	N
\CapThis	8	Gregory the Great . .	\Name 7
Carnap, Rudolph 13	 <i>see</i> Gregory I	\Name* 7
Charles V, emperor 8			\NamesActive 13
Charles the Bald, em-		H	\NamesFormat 12
peror 7, 8		Hammerstein, Oskar, II 9	\NamesInactive 13
Chiang Kai-shek 9		Harnack, Adolf 21	
Cicero, M.T. 7, 8		Henry VIII, English	O
Clemens, Samuel L. . .		king 9, 20	Oberdiek, Heiko 2
. <i>see</i> Twain, Mark		Hope, Bob 15	
Confucius 7, 8		Hope, Leslie Townes .	P
	 <i>see</i> Hope, Bob	Patton, George S., Jr. 20
			Plato 19
		I	\PName 16
		\IndexActive 18	Ptolemy I Soter, king
		\IndexInactive 18 19, 20
		\IndexName 16	Public, J.Q. 19
		J	R
		Jean sans Peur, duke . . 14	Rambam 16,
		Jean the Fearless <i>see also</i> Maimonides
	 <i>see</i> Jean sans Peur	Rock, J.D., III 8
		K	S
		King, Martin Luther,	Schlicht, Robert 2, 11
		Jr. 10	\ShowComma 9
			Smith, John 5, 18
		L	Smith, John, the other
		Leo I, pope 17	one 18
		Leo the Great <i>see</i> Leo I	Smith, John, the third
		Louis XIV, French king 18, 19
	 10, 15	Snel van Royen,
		Louis the Pious, em-	Rudolph 16
		peror 19	Snel van Royen, Wille-
		Lueck, Uwe 2	brord 16
		Luecking, Dan 11	Snellius <i>see</i> Snel van
			Royen, Rudolph,
		M <i>see</i> Snel van
		Maimonides 16	Royen, Willebrord
E			
Einstein, Albert 7, 8			
Eliot, George 15			
Evans, Mary Anne . .			
. <i>see</i> Eliot, George			
\ExcludeName	18		
F			
\FName	8		
\FName*	8		
\ForgetName	14		
Francis I, French king 20			

Stephani, Philipp	2	Thomas Aquinas	14	Vlad III Dracula	
Strietelmeier, John	9	Thomas of Aquino <i>see</i>		Vlad II Dracul	13
<code>\SubvertName</code>	<i>14</i>	Thomas Aquinas		Vlad II Dracul	12
Sullenberger, Chesley		Twain, Mark	16	Vlad III Dracula	13
B., III	8			Vlad III Dracula	13
Sun Yat-sen	10, 20	U		Vlad III Dracula	12, 13
Sun King . <i>see</i> Louis XIV		<code>\UntagName</code>	<i>17</i>	Vlad the Impaler <i>see</i>	
		V		Vlad III Dracula	
T		Vlad Țepeș	<i>see</i>	Voltaire	16
<code>\TagName</code>	<i>17</i>				