The nameauth package

Charles P. Schaum
charles dot schaum at comcast dot net

v1.9 from 2015/07/09

Abstract

The nameauth package automates the formatting and indexing of names.[1] This aids the use of a **name authority** and the process of textual reordering and revision without needing to retype name references.

Contents

[1] I omit the "Implementation" section to show the naming macros with a "normal" index.

1 Introduction

1.1 Preliminaries

Books can reference hundreds of names. It takes time and money to check them. This package helps to format and index names consistently and automatically, helping you save time and money. Features include:

- Simultaneously format, display, and index names.
- Change the syntactic format, typographical display, and other name features without retyping names.
- Display first and subsequent uses of names for professional writing. Default is a full name for the first use and shorter forms thereafter.
- Apply various custom typesetting formats to fit your needs without retyping names.
- Move text without retyping names.
- Show name variants in the text, yet index consistent name forms.
- Process different names according to cross-cultural naming conventions. These include mononyms, names with epithets, names with particles, and Eastern names in addition to basic Western names.
- Allow for different capitalizing and other conventions.
- Index different people with the same name by using a tag feature.
- Process names in list environments and other special environments.

I started using LaTeX and wrote this package for translating old German and Latin texts. Only in recent years have WYSIWYG apps become useful here. I learned much more than I expected regarding different LaTeX typesetting engines, different font systems, different indexing programs, different package interactions, and different naming conventions. In order to meet my cross-cultural needs amid such challenges, I encountered some technical restrictions:

Please avoid using control sequences like `\protected@edef` and `\noexpand` in names because this package uses `\protected@edef` as it parses its input (cf. Section 2.5.4). This is done in order to get name display and indexing to work under the constraints mentioned above. I advise the casual user not to perform a "torture test" on this package like I do in this manual.

This package depends on etoolbox, suffix, trimspaces, and xargs. It has been tested with latex, lualatex, pdflatex and xelatex. It will work with makeindex and texindy. This manual was typeset with pdflatex and makeindex.

Indexing generally conforms to the standard in Nancy C. Mulvany, *Indexing Books* (Chicago: University of Chicago Press, 1994). This should be suitable for a very wide application across a number of disciplines.

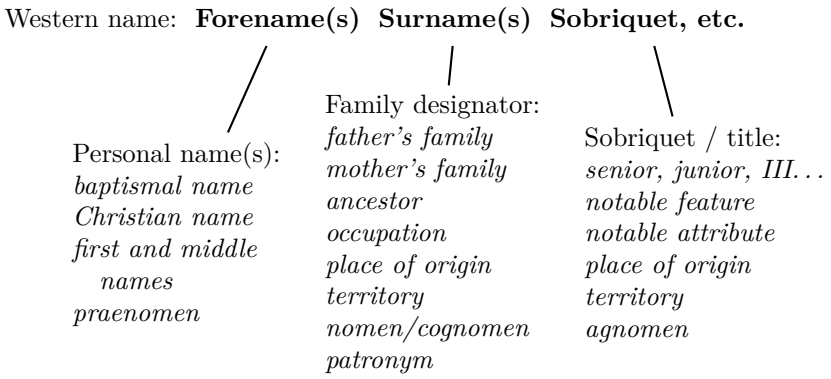
This documentation uses names of living and historical figures because users refer to real people in their projects. At no time do I intending any statement of bias for or against a particular person, culture, or tradition. All names mentioned herein deserve respect for the impact and legacy of their bearers.

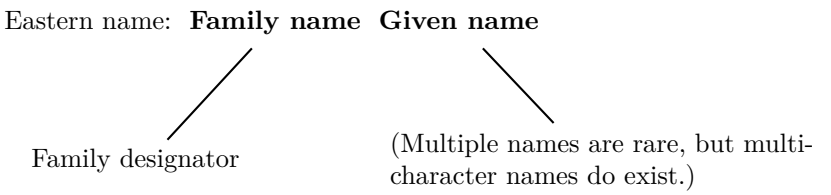
Thanks to Marc van Dongen, Enrico Gregorio, Philipp Stephani, Heiko Oberdiek, Uwe Lueck, and Robert Schlicht for their assistance.

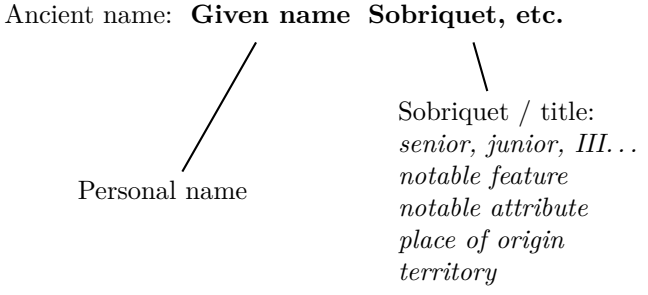
1.2 What's In A Name?

Here a name denotes a person. Apart from historical and cultural contexts, name forms are ambiguous. The nameauth package uses Boolean switches and macros to show names in specific contexts without the need to retype the names.

We see three categories of names, shown below. For special cases, one might have to decide how to handle Hungarian and Icelandic names as Eastern or Western. Professional writing often calls for the full form of a person's name to be used in its first occurrence, with shorter forms used thereafter. Some publications call for the first use of a name to be typeset differently than subsequent uses. The subsequent package options set broad parameters for such needs.







2 Usage

2.1 Package Options

Package options are designed to address how one expects name data to be entered, how names are displayed both semantically and typographically, and how they are indexed in different document sections. **Default options are in boldface**:

Show/Hide Affix Commas

`nocomma`	**Suppress commas between surnames and affixes, following the *Chicago Manual of Style* and other conventions.**
`comma`	Retain commas between surnames and affixes.

This option is set only at load time. The default `nocomma` option works better with Eastern names and modern standards, *e.g.*, James Earl Carter Jr. The `comma` option works with Western names and older standards, putting a comma before affixes, *e.g.*, James Earl Carter, Jr. One can switch options without retyping names and force commas with `\ShowComma` (Section 2.4.1).

Before version 0.9 the `nameauth` package assumed the `comma` option and its limits on `\AKA` and `\PName`. The older syntax remains only for backward compatibility.

Enable/Disable Formatting

`mainmatter`	**Enable typographic formatting attributes (see formatting options below), starting at the beginning of a document.**
`frontmatter`	Disable typographic formatting, but retain automatic full and short forms. See Section 2.5.6.

The default starts formatting names immediately. The `frontmatter` option is equivalent to `\NamesInactive`. It disables typographic name formatting until formatting is enabled with `\NamesActive`. It has no effect on syntactic name formatting, *i.e.*, the use of longer or shorter names depending on the context. It is designed for material provided by editors and contributors.

Enable/Disable Indexing

`index`	**Create index entries in place with names.**
`noindex`	Suppress indexing of names. See Section 2.7.5.

The default starts indexing names right away. The `noindex` option is equivalent to `\IndexInactive`. It disables the indexing of names until `\IndexActive` enables it. This applies only to naming and indexing macros in the `nameauth` package.

Capitalize Entire Surnames

`normalcaps`	**Do not perform any special capitalization.**
`allcaps`	Capitalize entire surnames, *e.g.*, romanized Eastern names. See Section 2.4.2.

This only affects the body text. To get caps in both the body text and the index, the user should type in the caps manually. Section-level control of family name capitalization occurs with `\AllCapsActive` and `\AllCapsInactive`. `\CapName` has the same effect on the individual instance of a name.

Reverse Name Order

`notreversed`	**Print names in the order specified by `\Name` and the other macros.**
`allreversed`	Print name forms in "smart" reverse order. See Sections 2.4.1 and 2.4.2.
`allrevcomma`	Print all names in Western "Surname, Forenames" order. See Section 2.5.1.

`\ReverseActive` and `\ReverseInactive` are used for reversing name order at the section level. `\ReverseCommaActive` and `\ReverseCommaInactive` can be used to create lists for educators and officials in the Western-style last-comma-first order, which is *not* the same as the `comma` option or `\ShowComma`. For individual instances of names, `\RevName` and `\RevComma` have the same respective effects.

Formatting Attributes

`alwaysformat`	If formatting is enabled by the `mainmatter` option or by `\NamesActive`, this option causes all names to have special typographic formatting.
`smallcaps`	**Set the first use of a name in small caps.**
`italic`	Italicize the first occurrence of a name.
`boldface`	Set the first use of a name in boldface.
`noformat`	Do not define a default format. Can be used with custom formatting. See Section 2.5.5.

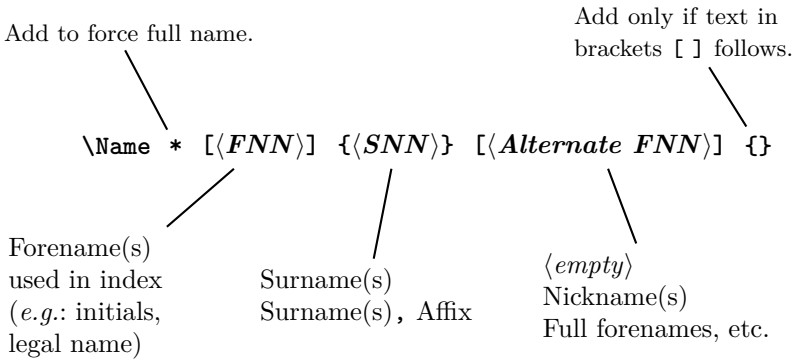
The default is `smallcaps` because this package was developed to aid the editing and translation of older German and Latin documents.

2.2 Quick Start Guide

2.2.1 Main Interface

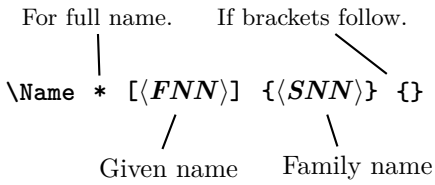
`\Name` The main workhorse of this package is `\Name`. More detail begins in Section 2.3. This overview shows how to work with the classes of names in Section 1.2. We abbreviate the command parameters ⟨*forename(s)*⟩ with ⟨*FNN*⟩ and ⟨*surname(s)*⟩ with ⟨*SNN*⟩. Note again that the `nocomma` option generally works best, especially with Eastern, medieval, and ancient names.

Western Names



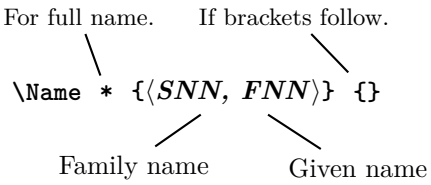
Use `\Name[`⟨*FNN*⟩`]{`⟨*SNN*⟩`}` and `\Name[`⟨*FNN*⟩`]{`⟨*SNN, Affix*⟩`}`. For nicknames and alternate forenames use `\Name[`⟨*FNN*⟩`]{`⟨*SNN*⟩`}[`⟨*Alternate FNN*⟩`]` and `\Name[`⟨*FNN*⟩`]{`⟨*SNN, Affix*⟩`}[`⟨*Alternate FNN*⟩`]`. The ⟨*Alternate FNN*⟩ are swapped with the non-empty ⟨*FNN*⟩, but only in the body text. The older, obsolete syntax is `\Name{`⟨*SNN*⟩`}[`⟨*Affix*⟩`]`.

Eastern Names in the Text, Western-style Index



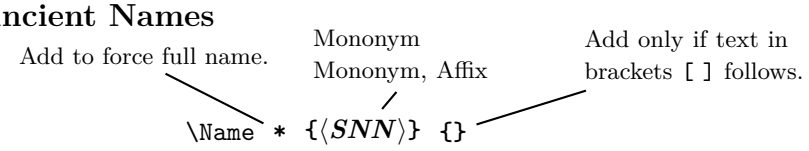
One uses the `allreversed` option or `\ReverseActive` or `\RevName` as appropriate (see above). The names in the body text will be in Eastern order, while they are indexed in Western fashion: ⟨*SNN*⟩, ⟨*FNN*⟩. This is the "non-native" form of Eastern names.

Eastern Names in the Text, Eastern-style Index



These names names will appear in the index using the Eastern form ⟨*SNN FNN*⟩ even if `\RevName` and friends are used to create a Western-order name in the body text. This is the "native" form of Eastern names. The older, obsolete syntax is `\Name{`⟨*SNN*⟩`}[`⟨*FNN*⟩`]`

Ancient Names



Use the forms `\Name{`⟨*Mononym*⟩`}` or `\Name{`⟨*Mononym, Affix*⟩`}`. The older, obsolete syntax is `\Name{`⟨*Mononym*⟩`}[`⟨*Affix*⟩`]`.

2.2.2 Simplefied Interface

`nameauth` The `nameauth` environment saves typing via shorthands. It is best used in the document preamble. Otherwise one must add `%` at the end of each line.

```
\begin{nameauth}
  \< ⟨cseq1⟩ & ⟨arg1⟩ & ⟨arg2⟩ & ⟨arg3⟩ >
  \< ⟨cseq2⟩ & ⟨arg1⟩ & ⟨arg2⟩ & ⟨arg3⟩ > ...
\end{nameauth}
```

Each ⟨*cseq*⟩ creates three macros. `\`⟨*cseq*⟩ itself calls `\Name`. `\L`⟨*cseq*⟩ (think "Long") calls `\Name*`. `\S`⟨*cseq*⟩ (think "Short") calls `\FName`. In the text, include trailing braces `{}` if text in brackets `[]` follows. Leading and trailing spaces in each field are stripped. There *must* be four argument fields per line:

```
\begin{nameauth}
  \< Wash & George & Washington & >                  Western
  \< Soto & Hernando & de Soto & >                   particle
  \< JRock & John David & Rockefeller, II & >         affix
  \< JayR & John David & Rockefeller, IV & Jay >     nickname[2]
  \< Aris & & Aristotle & >                          ancient
  \< Eliz & & Elizabeth, I & >                       royal
  \< Konoe & Fumimaro & Konoe & >                    Eastern/Western index
  \< Yamt & & Yamamoto, Isoroku & >                  Eastern/Eastern index
\end{nameauth}
```

`\Wash`: George Washington
`\Wash`: Washington
`\LWash`: George Washington
`\SWash`: George
`\RevComma\LWash`: Washington, George
`\Soto`: Hernando de Soto
`\Soto`: de Soto
`\CapThis\Soto`: De Soto
`\JRock`: John David Rockefeller II
`\JRock`: Rockefeller
`\JayR`: Jay Rockefeller IV
`\SJayR\ \JayR`: Jay Rockefeller

`\Aris`: Aristotle
`\Eliz`: Elizabeth I
`\Eliz`: Elizabeth
`\Konoe`: Fumimaro Konoe
`\RevName\LKonoe`: Konoe Fumimaro
`\CapName\LKonoe`: Fumimaro KONOE
`\CapName\RevName\LKonoe`: KONOE Fumimaro
`\Yamt`: Yamamoto Isoroku
`\CapName\LYamt`: YAMAMOTO Isoroku
`\RevName\LYamt`: Isoroku Yamamoto

[2] Careful here! `\SJayR` always gives "Jay." See Section 2.3.2.

2.2.3 Older Syntax

An older form of syntax remains for backward compatibility with early versions, especially with the `comma` option. It lacks some of the error checking and robustness of the new syntax and limits the use of several macros. Never mix the older and newer syntax! For the sake of completeness we have:

```
\Name{Dagobert}[I]                          royal name
\Name{Yoshida}[Shigeru]                     Eastern name
\begin{nameauth}
  \< Dagb & & Dagobert & I >                 royal name
  \< Yosh & & Yoshida & Shigeru >            Eastern name
\end{nameauth}
```

\Dagb: Dagobert I
\Dagb: Dagobert
\LDagb: Dagobert I

\CapName\Yosh: YOSHIDA Shigeru
\CapName\RevName\LYosh: Shigeru YOSHIDA

- This older syntax could produce weird results. `\Name{Henry}[VIII]` prints "Henry VIII" and "Henry" as expected. Using `\Name[Henry]{VIII}` (an error) prints either "Henry VIII" or just "VIII." Both `\Name{Henry}[VIII]` and `\Name[Henry]{VIII}` produce the same "first-use" control sequence. `\Name[Henry]{VIII}[Tudor]` gives "Tudor VIII" and "VIII."

  The newer syntax fixes many of these problems. `\Name{Henry, VIII}` gives "Henry VIII" and "Henry." It produces unambiguous control sequences and index entries from those above. Avoid `\Name{Henry, VIII}[Tudor]`; this gives "Henry VIII Tudor" and "Henry" in the text and "Henry VIII" in the index. Instead of that, add "Tudor" manually to `\Name{Henry, VIII}` in the text and use a tag in the index (Section 2.7.2).

- Avoid mixing the older and newer syntax! `\Name{Mononym, Affix}` and `\Name{Mononym}[Affix]` are different names! You get both Mononym Affix and Mononym Affix. Index tags for `\Name*[Henry]{VIII}` will not work with `\Name{Henry, VIII}`, and vice versa.

- The older syntax will not work with some macros. From the film *Men in Black III*, `\AKA{Boris}[the Animal]{Just Boris}` fails. `\Name{Boris, the Animal} \AKA{Boris, the Animal}{Just Boris}` works, and you get Boris the Animal being "Just Boris."

2.2.4 Tips and Warnings

- Keep it simple! Avoid macros that you do not need.
- Use the simplified interface or check your use of braces and brackets with naming macros to avoid execution halting and errors like "Paragraph ended. . . " and "Missing ⟨*grouping token*⟩ inserted."
- For stage names, etc., try using, *e.g.*, `\Name[J.]{Kreskin}[The Amazing]{\AKA[J.]{Kreskin}[Joseph]{Kresge}}`. You get The Amazing Kreskin (Joseph Kresge), with "Kreskin, J." in the index. You must have initials or something in the first optional field for this to work, or else it will fail.
- Special cases like "Iron Mike" Tyson can be handled with the intricate use of `\ForgetName` and `\SubvertName`, but it is easier to use manual formatting along with `\IndexName`. Using `‘‘\AKA[Mike]{Tyson}{Iron Mike}’’` creates "Iron Mike" in the text and a *see*-type cross-reference to the main name in the index. See Section 2.6.1.
- Avoid spaces between initials in first names. Either use `\frenchspacing` or *consistently* put fractional spaces between them. See also Bringhurst's *Elements of Typographic Style*.
- One way to spot errors is to compare index entries with names in the body text. All macros that produce output also emit meaningful warnings. `\PName` produces warnings via `\Name` and `\AKA`.
- Not all warnings are created equal. For example, `\AKA` warns one if used multiple times for the same name. That may or may not be what one wants and is left to the author. Other warnings, especially in the case of indexing macros, indicate that the macro produced no output.

Warnings result from:

1. Using a cross-reference `[`⟨*alternate FNN*⟩`]{`⟨*alternate SNN*⟩`}[`⟨*alt. names*⟩`]` created by `\AKA` as a reference in `\Name`, `\FName`, and `\PName`.
2. Using a reference `[`⟨*FNN*⟩`]{`⟨*SNN*⟩`}[`⟨*Alternate names*⟩`]` 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.
7. Using `\Name`, `\FName`, `\PName`, and `\AKA` to refer to names and cross-references excluded by `\ExcludeName`.

2.3 Naming Macros

2.3.1 Surnames: \Name and \Name*

\Name
\Name*

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

\Name[⟨*FNN*⟩]{⟨*SNN*⟩}[⟨*Alternate names*⟩]
\Name*[⟨*FNN*⟩]{⟨*SNN*⟩}[⟨*Alternate names*⟩]

Here we see how the syntax works:

\Name[Albert]{Einstein}	Albert Einstein
\Name*[Albert]{Einstein}	Albert Einstein
\Name[Albert]{Einstein}	Einstein
\Name{Confucius}	Confucius
\Name*{Confucius}	Confucius
\Name{Confucius}	Confucius
\Name[M.T.]{Cicero}[Marcus Tullius]	Marcus Tullius Cicero
\Name*[M.T.]{Cicero}[Marcus Tullius]	Marcus Tullius Cicero
\Name[M.T.]{Cicero}[Marcus Tullius]	Cicero
\Name{Charles, the Bald}	Charles the Bald
\Name*{Charles, the Bald}	Charles the Bald
\Name{Charles, the Bald}	Charles

\Name displays and indexes names, as illustrated in Section 2.9. It always prints the ⟨*SNN*⟩ field. \Name prints the "full name" at the first occurrence, then the partial form thereafter. \Name* always prints the full name.

The ⟨*Alternate names*⟩ field replaces the ⟨*FNN*⟩ field in the text only if the ⟨*FNN*⟩ field is not empty; see the example for Cicero above. One can use a nickname in some instances while keeping the indexed form constant. Thus, \Name[M.T.]{Cicero}[Marcus Tullius] and \Name[M.T.]{Cicero} are equivalent, while \Name{Cicero}[Marcus Tullius] and \Name{Cicero} are not. If you are using the newer syntax, avoid forms like \Name{Charles}[the Bald].

```
\begin{nameauth}% In the preamble!
  \< Einstein & Albert & Einstein & >
  \< Cicero & M.T. & Cicero & >
  \< Confucius & & Confucius & >
  \< CBald & & Charles, the Bald & >
\end{nameauth}
```

Above we see the same setup with the new interface in the preamble. In the body text, \Einstein, \LEinstein, and \Einstein produce Albert Einstein, Albert Einstein, and Einstein. \CBald\ and \CBald give Charles the Bald and Charles. In the next section we shall see why \Cicero[Marcus Tullius] is preferred to get Marcus Tullius Cicero.

2.3.2 Forenames: \FName

\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 first reference to \FName always is a full name. That prevents a first-name-only reference before a person has been introduced. Intentionally, \FName *never* gives the first name with Eastern names. You must type the first name and then use \IndexName. For examples we see below:

\FName[Albert]{Einstein}	Albert Einstein
\FName[Albert]{Einstein}	Albert
\FName{Confucius}	Confucius
\FName{Confucius}	Confucius
\FName[M.T.]{Cicero}[Marcus Tullius]	Marcus Tullius Cicero
\FName[M.T.]{Cicero}[Marcus Tullius]	Marcus Tullius
\FName{Charles, the Bald}	Charles the Bald
\FName{Charles, the Bald}	Charles

Nicknames are used instead of the regular first names by including them in the ⟨*Alternate names*⟩ field. For example, when writing of aviation hero Chesley B. Sullenberger III use ``\FName[Chesley B.]{Sullenberger, III}[Sully]'' to get the nickname "Sully."

With the simplified interface, \SEinstein, \SConfucius, and \SCBald give Albert, Confucius, and Charles. You must choose whether you want the nickname or the formal name as the default. For example:

```
\begin{nameauth}
  \< Ches & Chesley B. & Sullenberger, III & >
  \< Sully & Chesley B. & Sullenberger, III & Sully >
\end{nameauth}
```

Subsequent references to \SChes produce "Chesley B." Subsequent references to \SSully produce "Sully." Now here it gets tricky: \SChes[Sully] gives "Sully" as expected. However, \SSully[Chesley B.] expands to what it says, not what you mean: \FName[Chesley B.]{Sullenberger, III}[Sully][Chesley B.]. That clearly gives you "Sully[Chesley B.]." Use the nickname field with caution!

2.3.3 Full Stop Detection

Affixes and initials could result in the period of an abbreviation like "Jr.," "Sr.," "d. J." (*der Jüngere*), and "d. Ä." (*der Ältere*) followed by the sentence full stop. These macros check for such collisions and drop the extra full stop as needed:

`\Name[Martin Luther]{King, Jr.}.`	Martin Luther King Jr.
`\Name[Martin Luther]{King, Jr.}.`	King.
`\Name[Martin Luther]{King, Jr.}`	King (*e.g.*, in a sentence)
`\Name*[Martin Luther]{King, Jr.}.`	Martin Luther King Jr.
`\Name*[Martin Luther]{King, Jr.}`	Martin Luther King Jr.

Full-stop detection also works with `\FName`. If a name reference is enclosed, *e.g.*, between grouping tokens, detection will not work.

2.4 Affixes and Eastern Names

2.4.1 Comma-Delimited Affixes

Comma-delimited affixes handle several different name types. The first is Western names with affixes. One must use `\Name[Oskar]{Hammerstein, II}` to get Oskar Hammerstein II and Hammerstein. *Always include a comma as an affix delimiter.* Extra spaces and trailing commas are ignored. Other name types include royal, medieval, and Eastern names:

`\Name{Louis, XIV}`	Louis XIV
`\Name{Louis, XIV}`	Louis
`\Name{Sun, Yat-sen}`	Sun Yat-sen
`\Name{Sun, Yat-sen}`	Sun

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 *see* reference from "Sun King" to "Louis XIV" in the index (Section 2.6.1).

`\KeepAffix` Put `\KeepAffix` before `\Name` or `\AKA` if a line break or page break divides a ⟨*SNN, affix*⟩ pair. This puts a non-breaking space between *SNN* and *affix* in the body text, but not in the index. Other options to fix bad breaks include using `\hbox`, kerning and spacing in the microtype package, etc.

`\ShowComma` The `comma` option is restrictive and used to reproduce older texts. `\ShowComma` gets the same results on a per-name basis while using the default `nocomma` option. With `\ShowComma\Name[Louis]{Gossett, Jr.}` one gets Louis Gossett, Jr. One must use `\ShowComma` consistently or risk errors in the body text and index.

Avoid using the older syntax, shown below. It does not handle Western names with affixes and some other name types. `\AKA` and `\PName` cannot create cross-references to these forms. Thus we have:

`\Name{Henry}[VIII]`	Henry VIII
`\Name{Henry}[VIII]`	Henry
`\Name{Chiang}[Kai-shek]`	Chiang Kai-shek
`\Name{Chiang}[Kai-shek]`	Chiang

These older forms work because no ⟨*FNN*⟩ are present. Otherwise you would get weird nicknames. Again, please avoid using the older syntax.

2.4.2 Eastern Names

The nameauth package offers two ways to handle romanized Eastern names. `\RevName\Name[`⟨*Eastern FNN*⟩`]{`⟨*Eastern SNN*⟩`}` and equivalents will produce an Eastern name in the body text and the Western form ⟨*SNN*⟩, ⟨*FNN*⟩ in the index, including the comma. One might call this "non-native" mode.

In contrast, both `\Name{`⟨*Eastern SNN, Eastern FNN*⟩`}` and the older syntax `\Name{`⟨*Eastern SNN*⟩`}[`⟨*Eastern FNN*⟩`]` produce an Eastern form ⟨*SNN*⟩ ⟨*FNN*⟩ in the body text and index without any comma. This can be called "native" mode. The goal for offering these two "modes" is to allow for greatest flexibility in indexing requirements, not favoring one over another.

`\ReverseActive` `\ReverseInactive` `\RevName`

The reverse output mechanism makes the "non-native" mode compatible with the "native" mode in the body text. One chooses the style of index entry desired, then uses the "native" or "non-native" approach as desired. In addition to the class options described in Section 2.1, `\ReverseActive` and `\ReverseInactive` toggle reversing on a larger scale, while `\RevName` is used once per `\Name`.

A list of Japanese music artists illustrates the use of `\RevName`. Both Aiko Nakano and Yoko Kanno are listed in Western order, then "non-native" Eastern order. The others are all listed in "native" Eastern mode, then reversed to Western order. Their index entries reflect that:

	unchanged	`\RevName`
`\Name*[Aiko]{Nakano}`	Aiko Nakano	Nakano Aiko
`\Name*{Arai, Akino}`	Arai Akino	Akino Arai
`\Name*{Ishida}[Yoko]`	Ishida Yoko	Yoko Ishida
`\Name*{Yohko}`	Yohko	Yohko

`\AllCapsActive` `\AllCapsInctive` `\CapName`

Use `\AllCapsActive`, `\AllCapsInactive`, and `\CapName` for fully-capitalized family names in the body text. These macros are analogous to the reversing macros above and may be used alone or with those macros, *e.g.* `\CapName\RevName\Name`:

	unchanged	`\CapName\RevName`
`\Name*[Yoko]{Kanno}`	Yoko KANNO	KANNO Yoko
`\Name*{Shikata, Akiko}`	SHIKATA Akiko	Akiko SHIKATA
`\Name*{Nogawa}[Sakura]`	NOGAWA Sakura	Sakura NOGAWA
`\Name*{Yohko}`	YOHKO	YOHKO

The reversing and capitalization macros also work with `\AKA`. They affect only the text, not the index. Whoever wants all-cap forms in the index will have to capitalize everything manually or modify the macros. Note also that the above examples include the older syntax. Remember that index entries for Aiko Nakano and Yoko Kanno are Western-form; the others are Eastern-form.

2.5 Other Naming Topics

2.5.1 Listing by Surname

\ReverseCommaActive
\ReverseCommaInactive
\RevComma

Another set of reversing macros, `\ReverseCommaActive`, `\ReverseCommaInactive`, and `\RevComma`, allows the easy generation of lists with surnames, followed by a comma, then forenames. The first two are broad toggles, while the third works on a per-name basis. Here is a good place to show incompatibility between Eastern, medieval, and royal names on the one hand and Western names on the other. An indiscriminate use of `\RevComma\Name...` can yield:

John Stuart Mill	Mill, John Stuart	OK
Oskar Hammerstein II	Hammerstein II, Oskar	OK
John Eriugena	Eriugena John	medieval incompatible
Mao Tse-tung	Tse-tung Mao	Eastern inconpatible
Anaximander	Anaximander	OK

It is not possible for this package to be all things to all names, but it tries in good faith to be as cross-cultural as possible.

2.5.2 Naming Standards

\CapThis

According to the *Chicago Manual of Style*, 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'* always 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 *de* when it arises at the beginning of a sentence, use `\CapThis\Name[Walter]{de la Mare}`. De la Mare will think it fair. Non-English contexts do not always bind particles to surnames. Using `\FName` with alternate forenames allows greater flexibility here. See Section 2.9.2.

2.5.3 Hints for babel

Hyphens

The simplified interface trivializes the consistent insertion of optional hyphens in names arguments. Name hyphenation also can be aided by using the babel or polyglossia packages. With English as the main language, a name of German origin might break poorly. For example, John Strietelmeier might break thus: "Strietelmeier." Using `\newcommand{\de}[1]{\foreignlanguage{ngerman}{#1}}` and `\de{\Name[John]{Strietelmeier}}` (with babel) prevents such breaks.

texindy

Using babel with Roman page numbers will put `\textlatin` in the index entries if one includes a language that does not use the Latin alphabet — even if the main language does. The texindy program will ignore all such references. This issue can affect nameauth. One workaround for texindy could enclose text with any macros that write to the index in an environment or a `\long` macro defined like:

```
\newcommand{\fix}[1]{\def\textlatin##1{##1}#1}
```

2.5.4 Accented Names

Using xindy (texindy) and xelatex or lualatex is recommended for accented names. Under NFSS these Unicode characters are available (with 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 ă Ą ą Ć ć Č č	Ď ď Đ đ Ę ę Ě ě	Ğ ğ İ ı	FIRST USE
Ă ă Ą ą Ć ć Č č	Ď ď Đ đ Ę ę Ě ě	Ğ ğ İ ı	second use
IJ ıj Ľ ŀ Ł ł	Ń ń Ň ň Œ œ	Ŕ ŕ Ř ř	FIRST USE
IJ ij Ľ ľ Ł ł	Ń ń Ň ň Œ œ	Ŕ ŕ Ř ř	second use
Ś Ş ş Š š Ţ ţ Ť ť	Ů ů Ű ű	Ź ź Ż ż Ž ž	FIRST USE
Ś Ş ş Š š Ţ ţ Ť ť	Ů ů Ű ű	Ź ź Ż ż Ž ž	second use

One may use expandable control sequences in names (thanks Robert Schlicht). Protected or unexpandable control sequences via `\protected@edef` or `\noexpand` may generate, respectively, empty index entries or unbalanced groups/erroneous entries in the auxiliary files. Thanks to Patrick Cousot for pointing this out.

This cannot be changed because nameauth uses `\protected@edef` when it parses the name parameters. This is needed to work with makeindex, texindy, microtype, and other packages and LaTeX engines.

Additional accents and glyphs can be used with Unicode input, NFSS, inputenc, and fontenc when using fonts with TS1 glyphs, *e.g.*, `\usepackage{lmodern}` (per the table on pages 455–63 in *The Latex Companion*):

```
\usepackage{newunicodechar}
\DeclareTextSymbolDefault{\textlongs}{TS1}
\DeclareTextSymbol{\textlongs}{TS1}{115}
\newunicodechar{ſ}{\textlongs}
```

That lets you type "In Congreſs, July 4, 1776." Similarly, the following allows `\Name{Ghazāli}` to generate Ghazāli: `\newunicodechar{ā}{\=a}`

To get proper sorting with accents and `makeindex -g`, consider creating your own `.ist` file (pages 659–65 in *The Latex Companion*) or the following form: `\IndexInactive\Name{`⟨*actual*⟩`}\index{`⟨*sortkey*⟩`@`⟨*actual*⟩`}\IndexActive.`

Control sequences like `\=a` fail when using makeindex and `gind.ist` because the equal sign is an "actual" character instead of `@`. `\index{Gh{\=a}zali}` halts execution. `\index{Gh\=azali}` gives an "azali" entry sorted under "Gh" (thanks Dan Luecking). This issue is not specific to nameauth.

This package tries to work with multiple languages and typesetting engines. The following preamble snippet illustrates how that can be done:

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

The following can be used 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 ⟨pdflatex text⟩%
    \else ⟨latex text⟩%
    \fi%
  \fi%
\fi
```

2.5.5 Custom 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. Beyond 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 boldface, with additional notations in the margin if possible:

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

\Name{Vlad III, Dracula} was known as \AKA{Vlad III, Dracula}%
{Vlad}[Ţepeş], ``\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
Vlad II Dracul

Vlad III Dracula was known as Vlad Ţepeş, "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 `quote` environment permits local changes to `\NamesFormat` so they revert back to give: Vlad III Dracula and Vlad III. For references to "Vlad" one could use `\Name{Vlad, III Dracula}` instead. Do not mix `\Name{Vlad III, Dracula}` with `\Name{Vlad, III Dracula}` or the old syntax, lest errors bite! You would get multiple index entries with `\Name`, unwanted cross-references with `\AKA` and unexpected forms in the text. The simplified interface helps one to avoid this.

2.5.6 Disable Formatting

`\NamesActive`
`\NamesInactive`

Using the `frontmatter` option deactivates formatting until `\NamesActive` occurs. 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. Consider the two "species" of names to be "formatted" and "non-formatted," intended for independent sections of the document like front matter and main matter.

2.5.7 Tweaks: \ForgetName and \SubvertName

Perhaps the easiest way to avoid the "interspecies clashes" above are the two macros presented here. They 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. The next use of that name will print as if it were a "first use." Index entries and cross-references are *never* forgotten by this package.

\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 of that name will print as if it were a "subsequent use."

2.6 Name Variant Macros

2.6.1 \AKA

\AKA \AKA (meaning *also known as*) handles pseudonyms, stage names, *noms de plume*,
\AKA* and so on in order to replace typing manual cross-references in the index:

```
\Name{Jean, sans Peur} (\AKA{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.

Notice that "John the Fearless" receives no special formatting. This is intentional, as it reflects the idea of formatting only main index entries, not cross-references. Nevertheless, the reversing and capitalizing mechanisms do work with \AKA. The syntax for \AKA is:

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

Only the ⟨*FNN*⟩ and ⟨*SNN*⟩ arguments from \Name and friends may be cross-referenced. The new syntax allows \AKA to cross-reference all name types. Both macros create a cross-reference in the index from the ⟨*Alt. FNN*⟩, ⟨*Alt. SNN*⟩, and ⟨*Alt. names*⟩ fields to a name defined by ⟨*FNN*⟩ and ⟨*SNN*⟩, regardless of whether that name has been used. Both prevent double periods at the end of a sentence.

Both macros print only the ⟨*Alt. FNN*⟩ and ⟨*Alt. SNN*⟩ fields in the body text. If the ⟨*Alt. names*⟩ field is present, \AKA swaps ⟨*Alt. names*⟩ with ⟨*Alt. FNN*⟩

in the body text. `\AKA*` just prints ⟨*Alt. names*⟩ (if present) in the body text. Section 2.7.2 further illustrates `\AKA`, `\AKA*`, and index tagging.

For the following name types, `\AKA` and `\AKA*` yield the same results, using Bob Hope, Louis XIV, and Lao-tzu as examples:

`\AKA[Bob]{Hope}[Leslie Townes]{Hope}`	Leslie Townes Hope
`\AKA{Louis, XIV}{Sun King}`	Sun King
`\AKA{Louis}[XIV]{Sun King}`	FAIL
`\AKA{Lao-tzu}{Li, Er}`	Li Er

`\AKA` fails with the old syntax. One can use the ⟨*Alt. names*⟩ field with the names above, but that field does not generate a cross-reference. The ⟨*Alt. names*⟩ field (whis is a part of the cross-reference) was envisaged for names like Gregory I:

`\AKA{Gregory, I}{Gregory}[the Great]`	Gregory the Great
`\AKA*{Gregory, I}{Gregory}[the Great]`	the Great
`\AKA{Gregory}[I]{Gregory}[the Great]`	FAIL
`\AKA*{Gregory}[I]{Gregory}[the Great]`	FAIL

`\Name*{Gregory, I}` ``\AKA*{Gregory, I}{Gregory}[the Great]'' produces Gregory I "the Great" in the text and a *see* reference from Gregory the Great to Gregory I in the index. This may look like the old syntax, but it is not. The comma-delimited affix form in ⟨*Alt. SNN*⟩ is designed for Eastern names, while ⟨*Alt. names*⟩ is not. One can refer to Lafcadio Hearn with `\CapName\AKA[Lafcadio]{Hearn}{Koizumi, Yakumo}`: KOIZUMI Yakumo.

Using `\RevComma` before `\AKA[Bob]{Hope}[Leslie Townes]{Hope}` produces Hope, Leslie Townes, while `\RevName` omits the comma. Du Cange refers to Charles du Fresne: `\CapThis\AKA[Charles]{du Fresne}{du Cange}`. See also Section 2.9.2 on how to deal with different standards for particles.

Here we show `\AKA` used as a name in the running text:

```
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.

Cross-references generated by `\AKA` and `\AKA*` are meant only to be *see* references, never page entries. See also Section 2.2.4. In certain cases, the alternate name might need to be indexed with page numbers and *see also* references. Do not use `\AKA` in those cases, rather, consider the following:

Refer, *e.g.*, to `\Name{Maimonides} (\AKA{Maimonides}{Moses ben-Maimon)`: Maimonides (Moses ben-Maimon). That is a name and a *see* reference. Now one must refer to `\Name{Rambam}`: Rambam before making a cross-reference to Maimonides. Add `\index{Rambam|seealso{Maimonides}}` at the end of the document to ensure that it is the last entry.[3]

[3] Different standards exist for punctuating index entries and cross-references. Check with your publisher, style guide, docs for xindy and makeindex, and http://tex.stackexchange.com.

`\AKA` purposefully will not create multiple instances of a cross-reference. To deal with the special case where one moniker applies to multiple people, *e.g.*, "Snellius" for both Willebrord Snel van Royen and his son Rudolph Snel van Royen, use a manual solution:

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

The old syntax (`\Name{Jean}[sans Peur]`) does not work with `\AKA`. Even with the new syntax, using makeindex may require 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.

2.6.2 `\PName`

`\PName` `\PName` is a "convenience macro" meant for English-style Western names that sacrifices flexibility for simplicity. It generates a Western-style main name followed by a cross-reference in parentheses with the following syntax:

`\PName[`⟨*FNN*⟩`]{`⟨*SNN*⟩`}[`⟨*other FNN*⟩`]{`⟨*other SNN*⟩`}[`⟨*other alt.*⟩`]`

Although `\PName` creates an easy shortcut, its drawbacks are many. It only can use the ⟨*FNN*⟩⟨*SNN*⟩ form of `\AKA`. It cannot use `\AKA*`. `\PName` really is ill-suited to work with `\CapName`, `\CapThis`, `\RevComma`, `\RevName`, and the related package options. Use it if it is useful, and *caveat auctor*.

The author determines the name that is indexed (the first name) and the subsequent name that only occurs as a *see* reference. That subsequent name is never shortened in the text. To do that, using the table below, one would type, *e.g.*, `Arouet\IndexName{Voltaire}` or use the Rambam example above. `\PName` can generate the following examples:

`\PName[Mark]{Twain}[Samuel L.]{Clemens}`	Mark Twain (Samuel L. Clemens)
`\PName*[Mark]{Twain}[Samuel L.]{Clemens}`	Mark Twain (Samuel L. Clemens)
`\PName[Mark]{Twain}[Samuel L.]{Clemens}`	Twain (Samuel L. Clemens)
`\PName{Voltaire}[François-Marie]{Arouet}`	Voltaire (François-Marie Arouet)
`\PName*{Voltaire}[François-Marie]{Arouet}`	Voltaire (François-Marie Arouet)
`\PName{Voltaire}[François-Marie]{Arouet}`	Voltaire (François-Marie Arouet)

If one used `\PName{William, I}[William]{the Conqueror}` the body text would look right but the index cross-reference would be in error. Medieval and Eastern names are not suited for `\PName`. For them use `\AKA`.

2.7 Indexing Macros

2.7.1 \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[⟨*FNN*⟩]{⟨*SNN*⟩}[⟨*Alternate names*⟩]

\IndexName complies with the new syntax. If ⟨*FNN*⟩ are absent, it indexes ⟨*Alternate names*⟩ as an affix using the old syntax; otherwise it ignores ⟨*Alternate names*⟩. If indexing is switched off (see Section 2.7.5), this macro does nothing. It will not create index entries for names used with \AKA as cross-references.

The indexing mechanism in the nameauth package follows *Chicago Manual of Style* standards regarding Western names and affixes. Thus the name Chesley B. Sullenberger III becomes "Sullenberger, Chesley B., III" in the index. This formatting only occurs for Western names (where ⟨*FNN*⟩ are present).

2.7.2 \TagName

\TagName This macro creates an index tag that will appear in all entries for a corresponding \Name from the point of invocation until the end of the document or corresponding \UntagName. Both \TagName and \UntagName handle their arguments like \IndexName. Tag names in the preamble if possible. If used in the body to create or switch index tags, put a comment delimiter at the end of the macro.

\TagName[⟨*FNN*⟩]{⟨*SNN*⟩}[⟨*Alternate names*⟩]{⟨*tag*⟩}

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:

`\TagName{Leo, I}{, pope}`	Tag these names at the beginning
`\TagName{Gregory, I}{, pope}`	of the document.
...	
`\Name*{Leo, I}`	First references to Leo I and
`\Name*{Gregory, I}`	Gregory I
...	
`\Name*{Leo, I} was known as`	Leo I was known as Leo the
`\AKA{Leo, I}{Leo}[the Great].`	Great.
...	
`\Name{Gregory, I}`	Gregory "the Great" was another
``\AKA*{Gregory, I}%	major pope.
{Gregory}[the Great]''	

\TagName causes the index entries for Gregory I and Leo I to have the tag ",␣pope" added automatically. One must add a space manually at the front of a tag if one wants it. Otherwise one could add asterisks, daggers, and so on instead of a space and some text, sich as regnal or life dates, etc. The tag is literal text for

whatever you need. For example, all fictional names in the index of this manual have an asterisk without any spaces. Tagging aids scholarly indexing.

2.7.3 \UntagName

\UntagName

\TagName will replace one tag with another tag, but it does not remove a tag from a name. That is the role of \UntagName. The syntax is:

\UntagName[⟨*FNN*⟩]{⟨*SNN*⟩}[⟨*Alternate names*⟩]

By using \TagName and \UntagName, one can disambiguate different people with the same name. For example:

```
This refers to \Name*[John]{Smith}.
Now another \ForgetName[John]{Smith}%
\TagName[John]{Smith}{ (the other one)}\Name[John]{Smith}.
Then a third \ForgetName[John]{Smith}%
\TagName[John]{Smith}{ (the third)}\Name[John]{Smith}.
Then the first \UntagName[John]{Smith}\Name*[John]{Smith}.
```

This refers to John Smith.	*index*: Smith, John
Now another John Smith.	*index*: Smith, John (the other one)
Then a third John Smith.	*index*: Smith, John (the third)
Then the first John Smith.	*index*: Smith, John

The tweaking macros \ForgetName and \SubvertName make it seem like you are dealing with three people who have the same name. The index tags will group together those entries with the same tag.

Since this document puts an asterisk by all fictional names in the index, it puts an asterisk at the beginning of the tags above and does not \UntagName John Smith, but retags him with an asterisk again.

2.7.4 Global Name Exclusion

\ExcludeName

This macro globally prevents the indexing of a particular name or cross-reference. If you do not use it at the beginning of the document, you may not exclude any name or cross-reference that has been used already. The syntax is:

\ExcludeName[⟨*FNN*⟩]{⟨*SNN*⟩}[⟨*Alternate names*⟩]

For example, \ExcludeName[Kris]{Kringle} will permit Kris Kringle and Kringle to appear in the body text via \Name[Kris]{Kringle}, but no index entry can occur for this name. \ExcludeName[Santa]{Claus} will prevent \AKA[Kris]{Kringle}[Santa]{Claus} Santa Claus from generating a cross-reference. It is likelier that you would enclose \Name, etc. between \IndexInactive and \IndexActive (below) in order to suppress just one index entry.

2.7.5 Indexing Certain Sections

\IndexActive
\IndexInactive

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

2.8 Variant Spellings

Handling variant name spellings can be complicated. For example, one might settle on the form W.E.B. Du Bois in one's name authority. Yet an essay might use an alternate spelling for Du Bois, namely, 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. Use `\ForgetName` for the first use when spaces are the only variation between name forms because the mechanism for determining whether or not a name has occurred ignores spaces.
2. Wrap `\Name` and friends between `\IndexInactive` and `\IndexActive`.
3. Call `\IndexName` with the authoritative form right after `\IndexActive`.
4. Create a cross-reference in the index.

This can look like:

```
\gdef\DuBoisAlt{\IndexInactive\Name[W.E.B.]{DuBois}%
\IndexActive\IndexName[W.E.B.]{Du Bois}}
\index{DuBois, W.E.B.|see{Du Bois, W.E.B.}}...

...\ForgetName[W.E.B.]{DuBois}...\DuBoisAlt
```

The alternate section mentions W.E.B. DuBois, then just DuBois thereafter. The index will only hold the standard entry for W.E.B. Du Bois: "Du Bois, W.E.B." and a cross-reference from the variant "DuBois, W.E.B." to the standard entry.

2.9 Naming Pattern Reference

2.9.1 Basic Naming

Western Names

First reference in the text:
John Smith
	`\Name*[John]{Smith}`
	`\Name[John]{Smith}`
	`\FName[John]{Smith}`

Subsequent full name: John Smith	`\Name*[John]{Smith}`
Subsequent surname: Smith	`\Name[John]{Smith}`
Subsequent forename: John	`\FName[John]{Smith}`

Long first reference:
Jane Q. Public
	`\Name*[J.Q.]{Public}[Jane Q.]`
	`\Name[J.Q.]{Public}[Jane Q.]`
	`\FName[J.Q.]{Public}[Jane Q.]`

Subsequent full: J.Q. Public	`\Name*[J.Q.]{Public}`
Alternate forenames: Jane Qetsiyah Public	`\Name*[J.Q.]{Public}[Jane Qetsiyah]`
Alternate forename: Janie	`\FName[J.Q.]{Public}[Janie]`

Western Plus Affixes

Always use a comma to delimit name/affix pairs.

First reference:
George S. Patton Jr.
	`\Name*[George S.]{Patton, Jr.}`
	`\Name[George S.]{Patton, Jr.}`
	`\FName[George S.]{Patton, Jr.}`

Subsequent full: George S. Patton Jr.	`\Name*[George S.]{Patton, Jr.}`
Subsequent surname: Patton	`\Name[George S.]{Patton, Jr.}`
Subsequent forename: George	`\FName[George S.]{Patton, Jr.}[George]`

```
\begin{nameauth}% In the preamble!
  \< Smith & John & Smith & >
  \< JQP & J.Q. & Public & >
  \< Patton & George S. & Patton, Jr. & >
\end{nameauth}
```

`\Smith, \LSmith, \Smith, and \SSmith` give John Smith, John Smith, Smith, and John. `\JQP[Jane Q.], \LJQP[Jane Q.], and \JQP[Jane Q.]` make Jane Q. Public, Jane Q. Public, and Public. `\LJQP[Jane Qetsiyah]\ and \SJQP[Janie]` produce Jane Qetsiyah Public and Janie.

`\Patton, \LPatton, \Patton, and \SPatton` give George S. Patton Jr., George S. Patton Jr., Patton, and George S. For the alternate forename, `\SPatton[George]` yields George.

New Syntax: Royal, Eastern, and Ancient

Using `\Name{Demetrius, I Soter}` keeps the number with the affix. To keep the number with the name, use `\Name{Demetrius I, Soter}`. See also Section 2.4.1.

First reference: Francis I	`\Name*{Francis, I}`
	`\Name{Francis, I}`
	`\FName{Francis, I}`

Subsequent full: Francis I	`\Name*{Francis, I}`
Subsequent name: Francis	`\Name{Francis, I}` `\FName{Francis, I}`

First reference: Demetrius I Soter	`\Name*{Demetrius, I Soter}`
	`\Name{Demetrius, I Soter}`
	`\FName{Demetrius, I Soter}`

Subsequent full name: Demetrius I Soter	`\Name*{Demetrius, I Soter}`
Subsequent name: Demetrius	`\Name{Demetrius, I Soter}` `\FName{Demetrius, I Soter}`

First reference: Sun Yat-sen	`\Name*{Sun, Yat-sen}`
	`\Name{Sun, Yat-sen}`
	`\FName{Sun, Yat-sen}`

Subsequent full name: Sun Yat-sen	`\Name*{Sun, Yat-sen}`
Subsequent name: Sun	`\Name{Sun, Yat-sen}` `\FName{Sun, Yat-sen}`

First mononym reference: Plato	`\Name*{Plato}`
	`\Name{Plato}`
	`\FName{Plato}`

Subsequent mononym: Plato	`\Name*{Plato}`
	`\Name{Plato}`
	`\FName{Plato}`

```
\begin{nameauth}% In the preamble!
  \< Francis & & Francis, I & >
  \< Dem & & Demetrius, I Soter & >
  \< Sun & & Sun, Yat-sen & >
  \< Plato & & Plato & >
\end{nameauth}
```

`\Francis, \LFrancis, \Francis, and \SFrancis` give Francis I, Francis I, Francis, and Francis. `\Dem, \LDem, \Dem, and \SDem` make Demetrius I Soter, Demetrius I Soter, Demetrius, and Demetrius. `\Sun, \LSun, \Sun, and \SSun` produce Sun Yat-sen, Sun Yat-sen, Sun, and Sun. `\Plato, \LPlato, \Plato, and \SPlato` yield Plato, Plato, Plato, and Plato.

Note that `\KeepAffix\LFrancis` was used to prevent a bad break above. `\KeepAffix\Patton` prevented a bad break on the previous page.

Old Syntax: Royal and Eastern

Please note that these forms are not recommended. `\Name{Ptolemy}[I Soter]` keeps the number with the affix. To keep the number with the name, use `\Name{Ptolemy I}[Soter]`. See also Section 2.4.1.

First ref: HENRY VIII	`\Name*{Henry}[VIII]`
	`\Name{Henry}[VIII]`
	`\FName{Henry}[VIII]`

Subsequent refs: Henry VIII	`\Name*{Henry}[VIII]`
Subsequent refs: Henry	`\Name{Henry}[VIII]` `\FName{Henry}[VIII]`

First ref: PTOLEMY I SOTER	`\Name*{Ptolemy}[I Soter]`
	`\Name{Ptolemy}[I Soter]`
	`\FName{Ptolemy}[I Soter]`

Subsequent refs: Ptolemy I Soter	`\Name*{Ptolemy}[I Soter]`
Subsequent refs: Ptolemy	`\Name{Ptolemy}[I Soter]` `\FName{Ptolemy}[I Soter]`
First reference: MAO TSE-TUNG	`\Name*{Mao}[Tse-tung]` `\Name{Mao}[Tse-tung]`
Subsequent refs: Mao Tse-tung	`\Name*{Mao}[Tse-tung]`
Subsequent refs: Mao	`\Name{Mao}[Tse-tung]` `\FName{Mao}[Tse-tung]`

```
\begin{nameauth}% In the preamble!
  \< Henry & & Henry & VIII >
  \< Ptol & & Ptolemy & I Soter >
  \< Mao & & Mao & Tse-tung >
\end{nameauth}
```

`\Henry, \LHenry, \Henry, and \SHenry` give HENRY VIII, Henry VIII, Henry, and Henry. `\Ptol, \LPtol, \Ptol, and \SPtol` make PTOLEMY I SOTER, Ptolemy I Soter, Ptolemy, and Ptolemy. `\Mao, \LMao, \Mao, and \SMao` produce MAO TSE-TUNG, Mao Tse-tung, Mao, and Mao.

Avoid mixing syntax. If you take `\Name{Antiochus, IV}` and add a sobriquet like `\Name{Antiochus, IV}[Epiphanes]` you will get two different names in the index. `\AKA` may look like it blends old and new syntax, but it really does not.

Workarounds include `\Name{Antiochus, IV Epiphanes}` to get ANTIOCHUS IV EPIPHANES and Antiochus in the text and "Antiochus IV Epiphanes" in the index. Use `\Name{Antiochus~IV, Epiphanes}` to keep the numeral with the name (best done with the simplified interface). Maybe the best way is to use something like `\TagName{Antiochus, IV}{ Epiphanes, king}` with `\Name{Antiochus, IV}` and include "Epiphanes" manually as needed.

Remember also that you can "stack" modifiers like `\CapThis`, `\CapName`, `\RevName`, `\KeepAffix`, and so on in front of the control sequences of the simplified interface. `\CapName\LMao` generates MAO Tse-tung.

2.9.2 Particles

The following illustrate the American style of particulate names.

First: Walter de la Mare	`\Name*[Walter]{de la Mare}`
	`\Name[Walter]{de la Mare}`
	`\FName[Walter]{de la Mare}`

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

The Continental style differs slightly. These first three forms below put the particles in the index. Long macros are split for readability.

The (admittedly long) first use: Johann Wolfgang von Goethe	`\Name*[Johann Wolfgang von]{Goethe}`
	`\Name[Johann Wolfgang von]{Goethe}`
	`\FName[Johann Wolfgang von]{Goethe}`

Subsequent: Goethe	`\Name[Johann Wolfgang von]{Goethe}`
Forenames: Johann Wolfgang	`\FName[Johann Wolfgang von]{Goethe}%` `[Johann Wolfgang]`

These latter examples of the Continental style use the nickname feature to omit the particles from the index. Long macros are split for readability.

First: Adolf von Harnack	`\Name*[Adolf]{Harnack}[Adolf von]`
	`\Name[Adolf]{Harnack}[Adolf von]`
	`\FName[Adolf]{Harnack}[Adolf von]`

Next full name: Adolf von Harnack	`\Name*[Adolf]{Harnack}[Adolf von]`
Subsequent surname: Harnack	`\Name[Adolf]{Harnack}[Adolf von]` `\Name[Adolf]{Harnack}`
Subsequent forename: Adolf	`\FName[Adolf]{Harnack}`

```
\begin{nameauth}% In the preamble!
  \< DLM & Walter & de la Mare & >
  \< JWG & Johann Wolfgang von & Goethe & >
  \< Harnack & Adolf & Harnack & >
\end{nameauth}
```

`\DLM\ and \CapThis\DLM` produce Walter de la Mare and De la Mare. `\JWG\ and \JWG` give Johann Wolfgang von Goethe and Goethe. Using the alternate forenames option, `\Harnack[Adolf von]\ and \Harnack` yield Adolf von Harnack and Harnack, but you will not see Harnack's "von" in the index.

Change History

v0.7
General: Initial version 1
v0.75
General: Add new features. 1
v0.8
General: Improve functionality, compatibility, and docs. 1
v0.85
General: Add comma suppression, new options, more features. . . . 1
v0.86
General: Fix regressions. 1
v0.9
General: Add first name formatting; comma and affix handling expandable. 1
v0.94
General: Build with all major LaTeX engines. add index suppression, error checking, name particle caps. 1
v0.95
General: Bugfixes 1
v0.96
General: Bugfixes 1
v1.0
General: Works fully with microtype and memoir. 1
v1.1
General: Fix errors when emitting warnings. 1
v1.2
General: Add tagging features; improve documentation. 1
v1.26
General: Fix first-letter caps & sorting of name affixes in index. . . 1
v1.4
General: Fix moving arguments & other bugs; add features. 1
v1.5
General: Minor bugfixes; add features. 1
v1.7
General: Change options processing to prevent errors. 1
v1.8
General: Warn users to avoid protected accenting macros in names. 1
v1.9
General: Improve Eastern name support. Fix potential name collisions with the simplified interface. Ensure global undefining of names and tags. Rewrite documentation. 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.