

The nameauth package

Charles P. Schaum
`charles dot schaum at att dot net`

v0.86 from 2012/02/06

Abstract

These macros help an author to encode the (optional) forenames and (mandatory) surnames of people in order to normalize occurrences in the text and entries in the index. This package is intended for Western-style naming conventions.

1 Introduction

Suppose you were working on a collection of essays. Depending on the permissions governing the essays, you might need to allow variance in the spelling of people's names. You would use either a house reference, a research library, or an encyclopedic source to decide on authoritative name forms. Your index might use abbreviated versions of those forms. Either a substantive or copy editor would keep track of the main name form, the variants, and their relationship to the index. For optimal quality, that would require a proofreading pass with queries to the editor. That adds time and cost to the job. If the author or editor can handle these details in advance, the result trims overhead cost. That could make a prospective author or freelance editor more likely to be selected for publication.

1.1 Design Decisions

This package assumes that an author or editor wants to minimize keystrokes. The default behavior sets the first occurrence of a person's name in small caps and prints out that full name. Subsequent occurrences set the last name only in the running text font. Considerable variation from the defaults is possible.

1.2 Thanks

Thanks to MARC VAN DONGEN, ENRICO GREGORIO, PHILIPP STEPHANI, HEIKO OBERDIEK, and UWE LUECK for their invaluable assistance. Marc showed me the basic structure using the `xparse` package. Enrico and Philipp helped with generating control sequences and sanitizing. Heiko gave a space-removing solution that could be passed as an argument in a macro. Code adapted from Uwe's work on the `texhax` list enabled the routines to function with the `microtype` package.

2 Usage

2.1 Package Options

If the default behavior is not desired, the following options easily alter it. They are, as follows:

<code>mainmatter</code>	The default behavior triggers special typesetting of the first occurrence of a name, starting at the beginning of a document.
<code>frontmatter</code>	This option suppresses the special typesetting of the first occurrence of a name before the invocation of <code>\NamesActive</code> . This option fits well with front matter from a contributor who may not intend the same formatting and emphasis found in the main matter. The indexing and aliasing features of the package remain operative. <i>Note</i> : One can switch at will between formatted and non-formatted sections; see Section 2.7.
<code>smallcaps</code>	The default behavior when a name is first encountered is to print it in small caps.
<code>italic</code>	This option causes the first occurrence of the name to be italicized.
<code>boldface</code>	This option causes the first occurrence of the name to be set in boldface.
<code>noformat</code>	This option suppresses document formatting after the invocation of <code>\NamesActive</code> . If an author wants the indexing and aliasing functions without any special typesetting, this option accomplishes that easily. <i>Note</i> : One can override the name typesetting options manually; see Subsection 2.6.1. This option is <i>not</i> equivalent to <code>frontmatter</code> . The latter functions independently and never produces any formatting. This approach allows one to toggle formatting on and off without retyping the document. With <code>\NamesFormat</code> It anticipates variation among publishers' house styles.
<code>nocomma</code>	The default behavior suppresses printing of commas between surnames and suffixes, following modern styles like <i>Chicago Manual of Style</i> . See Subsection 2.4.1 for implications of this behavior.
<code>comma</code>	Print commas between surnames and suffixes, following older styles. See Subsection 2.4.1.

2.2 Quick Start Guide

This page and the next comprise a “cheat sheet” for the impatient. None of the examples in this subsection generate index entries. These examples simply help you get used to the package:

1. The indexed forms of the names always remain the same.
2. Always use the same form of reference, e.g., `\Name[John]{Smith}` or `\Name{Louis}[XIV]`, otherwise point 1 will become false.
3. Trade work for consistency. For example, `\Name[J.Q.]{Public}[Jane Q.]` can be moved around anywhere when you rearrange text and the first reference will always be the longer, optional form. `\Name[J.Q.]{Public}` might not do that.
4. Checking against the index will help you debug your `\Name` references, *inter alia*.
5. Start using the macros you *need*, then work from there.

I want to...	I need to use (for example)...
Print a full Western name (first reference)	<code>\Name*[John]{Smith}</code> or <code>\Name[John]{Smith}</code>
Full Western name	<code>\Name*[John]{Smith}</code>
Short Western name (subsequent reference)	<code>\Name[John]{Smith}</code>
Print a Western name that has a longer form in the text than in the index (first reference)	<code>\Name*[J.Q.]{Public}[Jane Q.]</code> or <code>\Name[J.Q.]{Public}[Jane Q.]</code>
Full name without longer form	<code>\Name*[J.Q.]{Public}</code>
Same person, last name only (subsequent reference)	<code>\Name[J.Q.]{Public}[Jane Q.]</code> or <code>†\Name[J.Q.]{Public}</code>
Print an ancient name	<code>\Name{Plato}</code> or <code>\Name*[Plato]</code>
Print a full Eastern name (first reference)	<code>\Name*[Mao]{Tse-tung}</code> or <code>\Name{Mao}[Tse-tung]</code>
Full Eastern name	<code>\Name*[Mao]{Tse-tung}</code>
Short Eastern name (subsequent reference)	<code>\Name{Mao}[Tse-tung]</code>
Print a full “royal” name (first reference)	<code>\Name*[Louis][XIV]</code> or <code>\Name{Louis}[XIV]</code>
Full “royal” name	<code>\Name*[Louis][XIV]</code>
Shorter subsequent reference	<code>\Name{Louis}[XIV]</code>
Ancient name and sobriquet	<code>\Name*[Antiochus V]{Eupator}</code> or <code>\Name{Antiochus V}[Eupator]</code>
Shorter subsequent reference	<code>\Name{Antiochus V}[Eupator]</code>

† This form is OK, but can cause unwanted results if you rearrange text.

I strongly suggest Subsection 2.4.1 for more on the interaction between suffixes and sobriquets. Especially these features create useful, yet ambiguous forms that may be hard to debug. Also, mind leading spaces in the macro arguments.

I want to...	I need to use (for example)...
Only index, not print a reference to “Public, J.Q.”	<code>\IndexName[J.Q.]{Public}</code> or <code>†\IndexName[J.Q.]{Public}[Jane Q.]</code>
Only index, not print a reference to “Plato”	<code>\IndexName{Plato}</code>
Only index, not print a reference to “Mao Tse-tung”	<code>‡\IndexName{Mao}[Tse-tung]</code>
Only index, not print a reference to “Louis XIV”	<code>‡\IndexName{Louis}[XIV]</code>
Only index, a reference to “Antiochus V Eupator”	<code>‡\IndexName{Antiochus V}[Eupator]</code>

† The alternate names are ignored in this case.

‡ Ambiguous variants are not shown. *Caveat auctor.*

I want to...	I need to use (for example)...
Print a full Western name followed by a lesser-known name in parentheses (first reference)	<code>\PName*[Bob]{Hope}[Leslie Townes]{Hope}</code> <code>\PName[Bob]{Hope}[Leslie Townes]{Hope}</code>
Full name, lesser-known name in parentheses	<code>\PName*[Bob]{Hope}[Leslie Townes]{Hope}</code>
Surname only, followed by a full lesser-known name (subsequent reference)	<code>\PName[Bob]{Hope}[Leslie Townes]{Hope}</code>
Mononym followed by a lesser-known name	<code>\PName*{Prince}[Prince Rogers]{Nelson}</code> <code>\PName{Prince}[Prince Rogers]{Nelson}</code>
Flexibly associate a name with a lesser-known name	...a tribute to <code>\Name{Ari Up}</code> , born <code>\AKA{Ari Up}[Arianne]{Forster}</code> ... One would not index <i>Ari Up</i> as “Up, Ari”.

Avoiding the following pitfalls will save time and frustration:

1. Mixing sobriquets with modern forms fails due to the “longer name” feature, e.g.,
`\Name{First}{Ancient}[Sobriquet]` and `\Name[King]{Number}[Sobriquet]`.
2. Using `\AKA` and `\PName` with “ancient,” “royal,” and “Eastern” forms will fail, but see Subsection 2.4.1 for a workaround:
`\Name{Ancient}[Sobriquet] \AKA{Ancient Sobriquet}[First]{Last}`
`\Name{Queen}[Number] \AKA{Queen Number}[First]{Last}`
`\Name{EastFamily}[EastFirst] \AKA{EastFamily EastFirst}[First]{Last}`
3. Using a forename or first initials can prevent failure in some cases:
OK: `\Name[J.]{Kreskin}[The Amazing] (\AKA[J.]{Kreskin}[Joseph]{Kresge})`
FAIL: `\Name[] {Kreskin}[The Amazing] (\AKA{Kreskin}[Joseph]{Kresge})`
FAIL: `\Name{Kreskin}[The Amazing] (\AKA{Kreskin}[Joseph]{Kresge})`
4. Keep track of how name references are disambiguated in the text. Since one must put the full indexed name in a `\Name` reference it is easy to forget that the reference will be shortened if it is subsequent. `\Name*` comes to the rescue here.
5. Mistyping the pairs of { braces } and [brackets] creates contextually-dependent errors that can be difficult to track. Leading spaces may be hard to spot in the text, but will be apparent in the index with incorrectly sorted names.

2.3 Basic Macros

2.3.1 `\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[\langle forename(s) \rangle]{\langle surname(s) \rangle}[\langle alternate names \rangle]
\Name*[\langle forename(s) \rangle]{\langle surname(s) \rangle}[\langle alternate names \rangle]
```

From now on we will abbreviate *forename(s)* with *FNN* and *surname(s)* with *SNN* at various points. The syntax descriptions above do not capture exactly how the `\Name` macro behaves. The following table should help:

<i>FNN</i>	<i>SNN</i>	Alternate Names	Result
Albert	Einstein	(none)	ALBERT EINSTEIN
(none)	Confucius	(none)	CONFUCIUS
M.T.	Cicero	Marcus Tullius	MARCUS TULLIUS CICERO
(none)	Charles	the Bald	CHARLES THE BALD

Basically, `\Name` connects the *FNN* to the *SNN* to create both a printed form and an indexed form, usually ***FNN SNN*** and ***SNN, FNN***. This takes care of most modern Western names. For those with one name, such as ancient figures or those with stage names, one can drop the *FNN* so that `\Name` produces the result ***SNN*** for both text and index.

A newer feature added to `\Name` avoids a complicated workaround. Sometimes you might want to have the option of using either an alternate set of forenames or a sobriquet that functions as a surname for ancient figures. These two alternatives are handled by the final, optional field of `\Name`. If “regular” *FNN* are present, then the alternate names conditionally will replace the *FNN* in the printed form, but not in the indexed form. If no regular *FNN* are present, then the alternate names will be appended to the *SNN* in the printed form *and* in the indexed form. You must use the sobriquet form of a name consistently or risk confusion.

I mentioned conditional use. The unstarred form prints the “full name” at the first occurrence, then only the partial form thereafter. The starred form always prints the full name. Both macros usually apply a different “font attribute” to the name when it first appears in the running text.

Here are some more examples of use:

```
\Name*[Johann Wolfgang von]{Goethe}
Print JOHANN WOLFGANG VON GOETHE the first time it appears.
Later, print the full name Johann Wolfgang von Goethe.

\Name[Johann Wolfgang von]{Goethe}
Print JOHANN WOLFGANG VON GOETHE the first time it appears.
Later, print only the last name Goethe.

\Name{Boethius} and \Name*{Boethius} Print BOETHIUS the first time it
appears. Later, print Boethius.
```

`\Name*{Antiochus IV}[Epiphanes]`

Print ANTIOCHUS IV EPIPHANES the first time it appears. Later, print the name and sobriquet Antiochus IV Epiphanes.

`\Name{Antiochus IV}[Epiphanes]`

Print ANTIOCHUS IV EPIPHANES the first time it appears. Later, print just the name Antiochus IV.

Note: Here I am playing a “dirty trick” by making each line begin as if the name had not yet occurred. See Section 2.8 for more.

2.3.2 Tabular Summary of `\Name`

Again, the common uses of `\Name` include:

<i>FNN</i>	<i>SNN</i>	Alternate Names	Result
Harry S.	Truman	(none)	HARRY S. TRUMAN
(none)	Mencius	(none)	MENCIUS
John Q.	Adams	John Quincy	JOHN QUINCY ADAMS
(none)	Ptolemy I	Soter	PTOLEMY I SOTER

Another option employs the sobriquet feature for royal names. The forms that work include:

<i>FNN</i>	<i>SNN</i>	Alternate Names	Result	Function
(none)	Henry	VIII	HENRY VIII	<code>\Name*{Henry}[VIII]</code>
(none)	Henry	VIII	Henry	<code>\Name{Henry}[VIII]</code>

Other options are possible and are discussed in Subsection 2.4.1. They are ambiguous and may not work in all circumstances, but they do afford compatibility with `\AKA` and `\PName`.

Rudimentary references to Eastern names (I plead ignorance on this point) can be achieved via the sobriquet feature:

<i>FNN</i>	<i>SNN</i>	Alternate Names	Result	Function
(none)	Chiang Kai-shek	Kai-shek	CHIANG KAI-SHEK	<code>\Name{Chiang}[Kai-shek]</code>
(none)	Chiang Kai-shek	Chiang		<code>\Name{Chiang}[Kai-shek]</code>

2.4 Advanced Topics

2.4.1 Suffix Removal

`\Name` and `\Name*` have an additional difference. `\Name` will truncate the suffixes from subsequent occurrences of last names. For example, it prints the name OSKAR HAMMERSTEIN II the first time and Hammerstein thereafter. *The typed form always uses a comma*, e.g., `\Name[Oskar]{Hammerstein, II}`. Moreover, the space after the comma is literal, but not manipulated by the package in case one wants to use a thin space instead. That is left to the author.

Again, **the comma is not optional with suffixes**. It is how this feature works. More than one comma in the arguments of `\Name` and all similar user interface macros will cause unwanted results.

`\Name` and friends 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 possible 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 class option `nocomma`, one can take advantage of the suffix feature to tweak more options out of `\Name`. Instead of the sobriquet feature, one could use the following variants:

<i>FNN</i>	<i>SNN</i>	Alternate Names	Result	Function
(none)	Louis, XIV	(none)	LOUIS XIV	<code>\Name{Louis, XIV}</code>
(none)	Louis, XIV	(none)	Louis	<code>\Name{Louis, XIV}</code>
(none)	Sun, Yat-sen	(none)	SUN YAT-SEN	<code>\Name{Sun, Yat-sen}</code>
(none)	Sun, Yat-sen	(none)	Sun	<code>\Name{Sun, Yat-sen}</code>

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 (see below). The sobriquet feature would otherwise prevent such usage.

Using this feature comes with a price. Even though suffix and sobriquet features produce the same *output* with the default options, they are internally *different* with respect to the first/subsequent use mechanism. Use each approach consistently. This workaround takes advantage of the ambiguous default behavior; the `comma` option will break its functionality. Again, *caveat auctor*.

2.4.2 Alternate forenames in the Text

I showed above the cases of Cicero and John Quincy Adams where one can have a longer name in the text than in the index. Here are fuller examples:

```

\Name[Wilhelm M.L.]{De Wette}[Wilhelm Martin Leberecht]
WILHELM MARTIN LEBERECHE DE WETTE (text, first occurrence)
De Wette (text, second occurrence)
The abbreviated name “De Wette, Wilhelm M.L.” is in the index.

\Name*[Wilhelm M.L.]{De Wette}
WILHELM M.L. DE WETTE (text, first occurrence)
Wilhelm M.L. De Wette (text, second occurrence)
The abbreviated name “De Wette, Wilhelm M.L.” is in the index.

```

```

\Name*[Wilhelm M.L.]{De Wette}[W.M.L.]
W.M.L. DE WETTE (text, first occurrence)
W.M.L. De Wette (text, second occurrence)
The abbreviated name “De Wette, Wilhelm M.L.” is in the index.

```

Alternate forenames *override* the *FNN* in the text. Sobriquets are *appended* to *SNN*. The presence or absence of *FNN* triggers this difference. Alternate forenames only get printed in subsequent occurrences of `\Name*`. The surname argument is *always* printed. The next macro allows one to bypass this restriction.

2.4.3 `\IndexName`

`\IndexName` This macro prints no text in the body. It only creates an index entry that conforms with the rest of the package. Its syntax is:

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

`\IndexName` *does not work* with the pen name mechanism described below. This example shows `\Name` and `\IndexName` working together.

```

\Name[C.F.W.]{Walther}[Carl Ferdinand Wilhelm] was a major
figure in American Lutheran history. \Name[C.F.W.]{Walther}
used his full name when signing documents. To family,
however, he was just Ferdinand\IndexName[C.F.W.]{Walther}.

CARL FERDINAND WILHELM WALTHER was a major figure in Ameri-
can Lutheran history. Walther used his full name when signing docu-
ments. To family, however, he was just Ferdinand.

```

One can use this macro for special cases, like referring to works by joint or multiple authors or simply indexing names consistently with this package:

```

\newcommand{\SJB}{%
{\IndexName[Stan]{Berenstain}\IndexName[Jan]{Berenstain}}
\textit{The Berenstain Bears} is a popular series of
children’s books with over 300 titles. One can index the
Berenstains\SJB{} in this manner.

```

The Berenstain Bears is a popular series of children’s books with over 300 titles. One can index the Berenstains in this manner.

```

First Maccabees recounts the suppression of Jewish customs
under the Seleucid king \Name{Antiochus IV}{Epiphanes}. His
behavior was so egregious that he became an early model for
the later profile of Antichrist.
\IndexName{Antiochus IV}{Epiphanes}

```

First Maccabees recounts the suppression of Jewish customs under the Seleucid king ANTIOCHUS IV EPIPHANES. His behavior was so egregious that he became an early model for the later profile of Antichrist.

2.4.4 Pen Names: An Introduction

`\IndexName` lacks some safeguards for handling pen names that the macros `\PName` and `\AKA` introduce. These macros are incompatible with the alternate name and sobriquet features of `\Name` for the following reasons:

1. It makes sense to have the better-known name come first.
2. That order produces an ambiguous argument list. Making the list unambiguous requires a mandatory argument, thus, extra keystrokes.
3. The more details in any given macro, the more confusing it is to use.
4. Many ancient pseudonyms require manual typesetting and indexing.

We already saw the strengths and weaknesses of the suffix feature above as a workaround for sobriquets and Eastern names when using `\PName` and `\AKA`. A more general workaround for such names follows:

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

JEAN SANS PEUR (Jean the Fearless) was Duke of Burgundy from 1404
to 1419.
```

A more complicated example is:

```
\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, so many modern scholars simply refer to
him as \Name{Thomas}[Aquinas].

Perhaps the greatest medieval theologian was THOMAS AQUINAS
(Thomas of Aquino), also known as Doctor Angelicus. His name
"Aquinas" is not a surname, so many modern scholars simply refer
to him as Thomas.
```

2.4.5 `\PName`

`\PName` Pseudonyms, stage names, *noms de plume*, epithets, and monikers can be handled with two macros: `\PName` and `\AKA`. The macro `\PName` usually takes a well-known name, followed by a lesser-known name. Like `\Name`, `\PName` has starred and unstarred forms, but those forms only operate on the first name, not on the second name. `\PName` also handles double-periods. The syntax is:

```
\PName[⟨FNN⟩]{⟨SNN⟩}[⟨lesser-known FNN⟩]{⟨lesser-known SNN⟩}
```

The following examples demonstrate uses of `\PName`. The macros are agnostic of the “pen name” itself. The author determines the name that is indexed (the first name) and the subsequent name that only occurs as a *see* reference.

```
\PName*[Mark]{Twain}[Samuel L.]{Clemens}
Print MARK TWAIN (Samuel L. Clemens) the first time it appears.
Later, print Mark Twain (Samuel L. Clemens).

\PName[Mark]{Twain}[Samuel L.]{Clemens}
Print MARK TWAIN (Samuel L. Clemens) the first time it appears.
Later, print Twain (Samuel L. Clemens).

\PName*[Willebrord]{Snel van Royen}{Snellius}
Print WILLEBRORD SNEL VAN ROYEN (Snellius) the first time it ap-
pears. Later, print Willebrord Snel van Royen (Snellius).

\PName[Willebrord]{Snel van Royen}{Snellius}
Print WILLEBRORD SNEL VAN ROYEN (Snellius) the first time it ap-
pears. Later, print Snel van Royen (Snellius).

\PName{Voltaire}[François-Marie]{Arouet}
Print VOLTAIRE (François-Marie Arouet) the first time it appears.
Later, print Voltaire (François-Marie Arouet). The starred version also
produces Voltaire (François-Marie Arouet)
```

2.4.6 \AKA

\AKA The second macro that handles aliases is `\AKA`. It is the same alias mechanism used in `\PName`, but separated for more flexibility. Its syntax is similar to `\PName`:

```
\AKA[⟨FNN⟩]{⟨SNN⟩}[⟨lesser-known FNN⟩]{⟨lesser-known SNN⟩}
```

`\AKA` only prints the pseudonym, *not* the indexed name. The macro also prevents double periods. An example of use follows:

```
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.
```

`\AKA` creates an index entry that cross-references to the “main” name given. It is assumed that a `\Name` macro occurs somewhere in reference to the indexed name. No error checking otherwise occurs for this.

Pseudonyms only work as cross-references. `\Name` will only print a pen name and emit a warning. To index a pseudonym with page numbers, use `\Name` for both the main name and the pseudonym, then manually cross-reference them, e.g.:

Authoritative Name	Alternate Name	Example of Use
MAIMONIDES	Moses ben-Maimon	<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 prevents bogus multiple cross-references in the index, but it also excludes 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). Do not use \IndexName in this case. One must add a manual index entry as a workaround:

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

Be careful with name collisions, especially with more than one person having the same last name. The following examples illustrate this:

```
\PName[Willebrord]{Snel van Royen}{Snellius}
Snel van Royen (Snellius)

\PName[Rudolph]{Snel van Royen}{Snellius}
Snel van Royen (Snellius)

\Name[Willebrord]{Snel van Royen}
Snel van Royen

\Name[Rudolph]{Snel van Royen}
Snel van Royen
```

2.5 Accented Names

The following Unicode accents will work in names using UTF8 and inputenc:

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
Ǻ ǻ Ǽ Ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	FIRST USE
Ǻ ǻ Ǽ Ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	second use
IJ iJ L l L l	Ń ń Ņ ņ Œ œ	Ř ř Ť ř	FIRST USE
IJ ij L l L l	Ń ń Ņ ņ Œ œ	Ř ř Ť 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

Other accents will not work unless you use T_EX control sequences or X_ƎL_AT_EX. You can also include the TS1 encoding and do something like the following with the inputenc package, the newunicodechar package, and and UTF8:

```

\DeclareTextSymbolDefault{\textlongS}{TS1}
\DeclareTextSymbol{\textlongS}{TS1}{115}
\newunicodechar{f}{\textlongS}
\newunicodechar{ā}{\=a}
\newunicodechar{ṁ}{\d{m}}

```

Please remember that the appropriate font packages, such as `lmodern` or the `TeX Gyre` fonts, are needed to obtain some TS1 glyphs. Also there may be points where `pdflatex` will accept the input (e.g., Ghāzali), but `makeindex` will have problems with that. In “normal” operation (I have used `article` and `memoir`) these control sequences appear to work fine. Special requirements like those in the `ltxdoc` class disrupt this functionality, requiring manual index entries in some cases.

2.6 Name Formatting

2.6.1 Font Attributes

`\NamesFormat` The first time a name is printed, it is formatted with the font attribute stored in `\NamesFormat`. This is set with the class options or manually. `\NamesFormat` can use either the command form or the declaration form of selecting font attributes, e.g., `\textsc` or `\scshape`. By redefining this macro, one can “hook” into the special typesetting of the first occurrence of a name. Consider the following:

```

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

```

If we `\let` the value of `\NamesFormat` to save the current value and implement a temporary change like the above, we get:

```

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

```

Vlad III Dracula
Vlad II Dracul

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

After using `\let` to revert `\NamesFormat`, a first occurrence again takes the form: VLAD III DRACULA.

Let me again stress that “royal” names used in this manner do not work with `\PName` and `\AKA`. Consistently use either the suffix mechanism (Subsection 2.4.1) or use as a guide the examples above for Jean sans Peur and Thomas Aquinas.

2.7 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 formatting on and off. The mechanism works in a complementary, yet independent manner. It can be used throughout the document.

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

<code>\Name[Rudolph]{Carnap}</code>	Rudolph Carnap
<code>\Name[Rudolph]{Carnap}</code>	Carnap
<code>\Name[Nicolas]{Malebranche}</code>	Nicolas Malebranche
<code>\Name[Nicolas]{Malebranche}</code>	Malebranche

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

<code>\Name[Rudolph]{Carnap}</code>	RUDOLPH CARNAP
<code>\Name[Rudolph]{Carnap}</code>	Carnap
<code>\Name[Nicolas]{Malebranche}</code>	NICOLAS MALEBRANCHE
<code>\Name[Nicolas]{Malebranche}</code>	Malebranche

2.8 Tweaks:\ForgetName

`\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 manner as `\Name`, 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 pseudonyms (see above) are *never* forgotten.

3 Implementation

3.1 Class Options and Required Packages

```
1 \newif\if@nameauth@DoFormat
2 \newif\if@nameauth@DoComma
3 \newif\if@nameauth@Punct
```

These Boolean values control or are controlled by formatting of first entries, suppression of commas, and whether or not a name ends with a period.

```
4 \newif\if@nameauth@dtx
```

This Boolean value handles proper generation of dtx files. It is set by `\@ifclassloaded`.

```
5 \DeclareOption{mainmatter}{\@nameauth@DoFormattrue}
6 \DeclareOption{frontmatter}{\@nameauth@DoFormatfalse}
7 \DeclareOption{smallcaps}{\newcommand{\NamesFormat}{\scshape}}
```

```

8 \DeclareOption{italic}{\renewcommand{\NamesFormat}{\itshape}}
9 \DeclareOption{boldface}{\renewcommand{\NamesFormat}{\bfseries}}
10 \DeclareOption{noformat}{\renewcommand{\NamesFormat}{}}
11 \DeclareOption{nocomma}{\@nameauth@DoCommafalse}
12 \DeclareOption{comma}{\@nameauth@DoCommatrue}
13 \ExecuteOptions{smallcaps,mainmatter,nocomma}
14 \ProcessOptions\relax
15 \RequirePackage{etoolbox}
16 \RequirePackage{xparse}
17 \@ifclassloaded{ltxdoc}
18   {\@nameauth@dtxttrue}
19   {\@nameauth@dtxfalse}%

```

The options above interact with the prior Boolean values. Suppressing and showing commas is set at load time and should not be changed in the document or else significant errors will result. `\@ifclassloaded` sets the Boolean value that governs the “literal” character used in the index. The other options can be changed without problem and several user interface macros provide that functionality.

3.2 Internal Macros

`\@nameauth@CleanName`

```
20 \newcommand*{\@nameauth@CleanName}[1]{\expandafter\zap@space\detokenize{#1} \@empty}
```

Thanks to Heiko Oberdiek, this macro produces a “sanitized” string based on the forename/surname parameters of `\Name` and friends. With this we can construct a control sequence name. Testing for the presence of that control sequence determines the existence of pseudonyms and the first occurrence of a name.

The following macros pertain to both the suffix mechanism mentioned above and the removal of suffixes with subsequent references of `\Name`.

`\@nameauth@Root`

```
21 \def\@nameauth@Root#1,#2*{#1}%
```

Strips off the second parameter delimited by a comma and an asterisk.

`\@nameauth@Suffix`

```

\@nameauth@Truncate 22 \def\@nameauth@Suffix#1,#2*{\ifx#2\@empty\else\@nameauth@Truncate#2\fi}
23 \def\@nameauth@Truncate#1,{#1}

```

Strips off the first parameter and cleans up any extra commas.

`\@nameauth@NoComma`

```

24 \newcommand{\@nameauth@NoComma}[1]{%
25   {\@nameauth@Root#1,*\@nameauth@Suffix#1,*}%
26 }

```

This is the heart of the comma suppression feature. Since the presence of commas are necessary for the suffix mechanism to work, commas are stripped as close as possible to generating output either for the body text or for the index.

`\@nameauth@Zapsuffix`

```
27 \newcommand*{\@nameauth@Zapsuffix}[1]{%
28     {\@nameauth@Root#1,*}%
29 }
```

Anything starting with a comma and ending with the end of the name is stripped off. That includes “Sr.,” “Jr.,” “III.,” and so on.

The following macros implement the mechanism to prevent the double-printing of a period after “Sr.,” “Jr.,” and so on.

`\@nameauth@CheckDot`

```
30 \def\@nameauth@CheckDot{\futurelet\@token\@nameauth@EvalDot}
```

This macro assigns the lookahead token `\@token` to be evaluated by `\@nameauth@EvalDot` while keeping `\@token` non-destructively on the list of input tokens. I use this method instead of `\@nextchar` because I do not want to gobble spaces.

`\@nameauth@EvalDot`

```
31 \def\@nameauth@EvalDot{\let\@period=. \ifx\@token\@period\expandafter\@gobble \fi}
```

`\@nameauth@EvalDot` checks if `\@token` is a period. If so it gobbles it by using `\expandafter` to get past the grouping. Another `\expandafter` occurs immediately before the invocation of `\@nameauth@CheckDot` in `\Name` and `\AKA`.

`\@nameauth@TestDot`

```
32 \newcommand*{\@nameauth@TestDot}[1]{%
33     \def\TestDot##1.\TestEnd##2\TestStop{\TestPunct{##2}}%
34     \def\TestPunct##1{\ifx\TestPunct##1\TestPunct\else\@nameauth@Puncttrue\fi}%
35     \@nameauth@Punctfalse%
36     \TestDot#1\TestEnd.\TestEnd\TestStop%
37 }
```

While `\@nameauth@CheckDot` looks *ahead* for a period, `\@nameauth@TestDot`—based on a snippet by Uwe Lueck—checks for a terminal period in the name passed to it, ignoring medial periods. It always resets the Boolean value before making its test, making it unnecessary to reset elsewhere.

`\@nameauth@FmtName`

```
38 \DeclareDocumentCommand\@nameauth@FmtName{s m}%
39 {%
40     \@nameauth@TestDot{#2}%
41     \if@nameauth@DoComma%
42         \IfBooleanTF{#1}{#2}{\bgroup\NamesFormat{#2}\egroup}%
43     }
```

```

43     \else
44         \IfBooleanTF{#1}%
45             {\@nameauth@NoComma{#2}}%
46             {\bgroup\NamesFormat{\@nameauth@NoComma{#2}}\egroup}%
47     \fi
48 }

```

`\@nameauth@FmtName` is where the first occurrences of a name are formatted. Notice how `\NamesFormat` sits between a `\bgroup` and an `\egroup` to localize the font change. The `\NamesFormat` hook has been discussed above. Comma suppression occurs here in order for the period-checking routines to work. That means comma suppression operates on the whole name here, but only on the *SNV* field for the index entries. This should not present a noticeable difference, since commas seldom if ever occur in forenames.

3.3 User Interface Macros

`\Name`

```

49 \DeclareDocumentCommand\Name{s o m o}%
50 {%
51     \if@nameauth@DoComma\edef\Surnames{#3}%
52     \else\edef\Surnames{\@nameauth@NoComma{#3}}\fi%
53     \IfValueTF{#2}%
54     {\IfValueTF{#4}{\edef\Forenames{#4}}{\edef\Forenames{#2}}%
55     \ifcsname\@nameauth@CleanName{#2#3!PN!}\endcsname%
56         \@nameauth@FmtName*{#2 #3}%
57         \PackageWarning{nameauth}%
58         {You cannot create a page reference from the pen name: #2 #3.}%
59     \else%
60         \if@nameauth@DoFormat%
61             \ifcsname\@nameauth@CleanName{#2#3!MN!}\endcsname%
62             \IfBooleanTF{#1}%
63                 {\@nameauth@FmtName*{\Forenames{ } #3}%
64                 \if@nameauth@dtx\index{#3=\Surnames, #2}%
65                 \else\index{#3@\Surnames, #2}\fi}%
66                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}%
67                 \if@nameauth@dtx\index{#3=\Surnames, #2}%
68                 \else\index{#3@\Surnames, #2}\fi}%
69             \else%
70                 \csgdef{\@nameauth@CleanName{#2#3!MN!}}{ }%
71                 \@nameauth@FmtName{\Forenames{ } #3}%
72                 \if@nameauth@dtx\index{#3=\Surnames, #2}%
73                 \else\index{#3@\Surnames, #2}\fi%
74             \fi%
75         \else%
76             \ifcsname\@nameauth@CleanName{#2#3!NF!}\endcsname%
77             \IfBooleanTF{#1}%
78                 {\@nameauth@FmtName*{\Forenames{ } #3}%
79                 \if@nameauth@dtx\index{#3=\Surnames, #2}%

```



```

80         \else\index{#3@\Surnames, #2}\fi}%
81     {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}%
82         \if@nameauth@dtx\index{#3=\Surnames, #2}%
83         \else\index{#3@\Surnames, #2}\fi}%
84     \else%
85         \csgdef{\@nameauth@CleanName{#2#3!NF!}}{}%
86         \@nameauth@FmtName*{\Forenames{} #3}%
87         \if@nameauth@dtx\index{#3=\Surnames, #2}%
88         \else\index{#3@\Surnames, #2}\fi%
89     \fi%
90 \fi%
91 \fi}%
92 {\IfValueTF{#4}
93     {\ifcsname\@nameauth@CleanName{#3#4!PN!}\endcsname%
94         \@nameauth@FmtName*{#3 #4}%
95         \PackageWarning{nameauth}%
96         {You cannot create a page reference from the pen name: #3 #4.}%
97     \else%
98         \if@nameauth@DoFormat%
99             \ifcsname\@nameauth@CleanName{#3#4!MN!}\endcsname%
100             \IfBooleanTF{#1}%
101                 {\@nameauth@FmtName*{#3 #4}%
102                     \if@nameauth@dtx\index{#3=\Surnames{} #4}%
103                     \else\index{#3@\Surnames{} #4}\fi}%
104                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}%
105                     \if@nameauth@dtx\index{#3=\Surnames{} #4}%
106                     \else\index{#3@\Surnames{} #4}\fi}%
107             \else%
108                 \csgdef{\@nameauth@CleanName{#3#4!MN!}}{}%
109                 \@nameauth@FmtName{#3 #4}%
110                 \if@nameauth@dtx\index{#3=\Surnames{} #4}%
111                 \else\index{#3@\Surnames{} #4}\fi%
112             \fi%
113         \else%
114             \ifcsname\@nameauth@CleanName{#3#4!NF!}\endcsname%
115             \IfBooleanTF{#1}%
116                 {\@nameauth@FmtName*{#3 #4}%
117                     \if@nameauth@dtx\index{#3=\Surnames{} #4}%
118                     \else\index{#3@\Surnames{} #4}\fi}%
119                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}%
120                     \if@nameauth@dtx\index{#3=\Surnames{} #4}%
121                     \else\index{#3@\Surnames{} #4}\fi}%
122             \else%
123                 \csgdef{\@nameauth@CleanName{#3#4!NF!}}{}%
124                 \@nameauth@FmtName*{#3 #4}%
125                 \if@nameauth@dtx\index{#3=\Surnames{} #4}%
126                 \else\index{#3@\Surnames{} #4}\fi%
127             \fi%
128         \fi%
129     \fi}%

```

```

130      {\ifcsname \@nameauth@CleanName{#3!PN!}\endcsname%
131        \@nameauth@FmtName*{#3}%
132        \PackageWarning{nameauth}%
133        {You cannot create a page reference from the pen name: #3.}%
134      \else%
135        \if@nameauth@DoFormat%
136          \ifcsname \@nameauth@CleanName{#3!MN!}\endcsname%
137          \IfBooleanTF{#1}%
138            {\@nameauth@FmtName*{#3}%
139              \if@nameauth@dtx\index{#3=\Surnames}%
140              \else\index{#3@\Surnames}\fi}%
141            {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}%
142              \if@nameauth@dtx\index{#3=\Surnames}%
143              \else\index{#3@\Surnames}\fi}%
144          \else%
145            \csgdef{\@nameauth@CleanName{#3!MN!}}{}}%
146            \@nameauth@FmtName{#3}%
147            \if@nameauth@dtx\index{#3=\Surnames}%
148            \else\index{#3@\Surnames}\fi%
149          \fi%
150        \else%
151          \ifcsname \@nameauth@CleanName{#3!NF!}\endcsname%
152          \IfBooleanTF{#1}%
153            {\@nameauth@FmtName*{#3}%
154              \if@nameauth@dtx\index{#3=\Surnames}%
155              \else\index{#3@\Surnames}\fi}%
156            {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}%
157              \if@nameauth@dtx\index{#3=\Surnames}%
158              \else\index{#3@\Surnames}\fi}%
159          \else%
160            \csgdef{\@nameauth@CleanName{#3!NF!}}{}}%
161            \@nameauth@FmtName*{#3}%
162            \if@nameauth@dtx\index{#3=\Surnames}%
163            \else\index{#3@\Surnames}\fi%
164          \fi%
165        \fi%
166      \fi}%
167    }%
168    \if@nameauth@Punct\expandafter\@nameauth@CheckDot\fi%
169  }

```

`\Name` is the heart of this package. Marc van Dongen provided the basic structure.

The reason why one sees six bifurcate outcomes repeated three times for eighteen “normal” and eighteen “special” variations is because one must make different decisions based on the dynamic interaction of many factors beyond mere input. First, at every instance where output is generated via `\index` there is a check to determine if the “normal” character for index entry literals will be used or the “special” version will be used with the `ltxdoc` class.

Regarding the many branches, one cannot assume that the name will terminate with a suffix like “Jr.” until a check has been run to determine if the suffix should be truncated. The main dynamic factors include: the Boolean and value parameters (there are eight valid patterns of input among them), the first or subsequent use, the front or main matter use, possible operations on suffixes, the formatting or non-formatting of output, and whether a pen name has been passed to the routine.

Here is how these many factors interrelate:

`\Name` first checks for comma suppression. if so it stores a comma-suppressed version of the third parameter in `\Surnames`. Otherwise it stores an unchanged version in `\Surnames`. This will be passed to `\index` as a literal.

`\Name` then checks for the forenames argument. Two outcomes are possible.

1. Forenames are present. In this case, the alternate names argument creates two choices.
 - (a) The alternate names replace the forenames in the printed form, not the indexed form.
 - (b) The absence of alternate names will result in the forenames being used for both forms.
2. Forenames are absent. In that case, the alternate names argument creates two different choices.
 - (a) The alternate names are appended to the surnames in both printed and indexed forms.
 - (b) Only the surnames are used.

A check occurs for a control sequence based on the names given and the suffix `!PN!` (for pen name). This is how `\AKA` protects cross-references. If the control sequence exists, then `\Name` only prints the name given (but with comma suppression) and emits a warning. One cannot use `\ForgetName` to expunge a pen name. This is a deliberate decision to avoid corruption of the index cross-references.

The next branch involves the Boolean value `@nameauth@DoFormat`, which is controlled by `\NamesActive` and `\NamesInactive`. If formatting is active, choose the unstarred form of `@nameauth@FmtName`, which applies the formatting hook. Otherwise use the starred form that applies no formatting.

The state of `@nameauth@DoFormat` also controls the suffix used in the control sequences: `!MN!` for main name or `!NF!` for no format. This is the heart of the `frontmatter` / `mainmatter` mechanism.

Where longer and shorter versions of names are printed, there the star parameter controls those outcomes. Comma suppression is used or not, as appropriate, in the index entries, requiring the use of the “literal” character `@` in most cases to get the alphabetizing and the output to work properly. Note again that one must always use commas with suffixes in the input, even if commas are suppressed in the output. That is how the mechanism works.

The use of `\expandafter` before `@nameauth@CheckDot` works with the other use of `\expandafter` mentioned with `@nameauth@EvalDot` above to move past

the closing brace and fetch the period as lookahead. That is only done when the check for a terminal period in the name succeeds.

`\PName`

```
170 \DeclareDocumentCommand\PName{s o m o m}%
171 {%
172     \IfBooleanTF{#1}{\Name*{#2}{#3}}{\Name{#2}{#3}}%
173     {\space}{\AKA{#2}{#3}{#4}{#5}}%
174 }
```

`\PName` is a convenience macro whose starred and unstarred forms call the respective versions of `\Name`, then `\AKA`. It prevents both the “sobriquet” feature and the alternate forenames feature, as mentioned above. It is only compatible with similar usage via `\Name` relying on the ambiguous default behavior of suffixes, the suffix suppression routines, and comma suppression.

`\AKA`

```
175 \DeclareDocumentCommand\AKA{o m o m}%
176 {%
177     \if@nameauth@DoComma%
178         \edef\Surnamesi{#2}\edef\Surnamesii{#4}%
179         \else\edef\Surnamesi{\@nameauth@NoComma{#2}}%
180         \edef\Surnamesii{\@nameauth@NoComma{#4}}%
181     \fi%
182     \IfValueTF{#3}%
183     {\@nameauth@FmtName*{#3 #4}%
184         \ifcsname\@nameauth@CleanName{#3#4!PN!}\endcsname\relax\else%
185             \csgdef{\@nameauth@CleanName{#3#4!PN!}}{}%
186             \IfValueTF{#1}%
187                 {\if@nameauth@dtx\index{#4=\Surnamesii, #3|see{\Surnamesi, #1}}%
188                     \else\index{#4@\Surnamesii, #3|see{\Surnamesi, #1}}\fi}%
189                 {\if@nameauth@dtx\index{#4=\Surnamesii, #3|see{\Surnamesi}}%
190                     \else\index{#4@\Surnamesii, #3|see{\Surnamesi}}\fi}%
191             \fi}%
192     {\@nameauth@FmtName*{#4}%
193         \ifcsname\@nameauth@CleanName{#4!PN!}\endcsname\relax\else%
194             \csgdef{\@nameauth@CleanName{#4!PN!}}{}%
195             \IfValueTF{#1}%
196                 {\if@nameauth@dtx\index{#4=\Surnamesii|see{\Surnamesi, #1}}%
197                     \else\index{#4@\Surnamesii|see{\Surnamesi, #1}}\fi}%
198                 {\if@nameauth@dtx\index{#4=\Surnamesii|see{\Surnamesi}}%
199                     \else\index{#4@\Surnamesii|see{\Surnamesi}}\fi}%
200             \fi}%
201     \if@nameauth@Punct\expandafter\@nameauth@CheckDot\fi%
202 }
```

`\AKA` prints a pseudonym and creates index cross-references. It also checks to see if cross-references already have been generated, and if so, it does not do it again. Like `\Name` it checks for a terminal period, but only for the alternate name.

`\IndexName`

```

203 \DeclareDocumentCommand\IndexName{o m o}%
204 {%
205     \if@nameauth@DoComma\edef\Surnames{#2}%
206     \else\edef\Surnames{\@nameauth@NoComma{#2}}\fi%
207     \IfValueTF{#1}%
208     {\ifcsname\@nameauth@CleanName{#1#2!PN!}\endcsname%
209     \else%
210         \if@nameauth@dtx\index{#2=\Surnames, #1}%
211         \else\index{#2@\Surnames, #1}\fi%
212     \fi}%
213     {\IfValueTF{#3}%
214     {\ifcsname\@nameauth@CleanName{#2#3!PN!}\endcsname%
215     \else%
216         \if@nameauth@dtx\index{#2=\Surnames{ } #3}%
217         \else\index{#2@\Surnames{ } #3}\fi%
218     \fi}%
219     {\ifcsname\@nameauth@CleanName{#2!PN!}\endcsname%
220     \else%
221         \if@nameauth@dtx\index{#2=\Surnames}%
222         \else\index{#2@\Surnames}\fi%
223     \fi}%
224     }%
225 }
```

`\IndexName` creates an index entry that is not already a pseudonym. It prints nothing. It does ensure consistent formatting.

`\ForgetName`

```

226 \DeclareDocumentCommand\ForgetName{o m o}%
227 {%
228     \IfValueTF{#1}%
229     {\csundef{\@nameauth@CleanName{#1#2!MN!}}}%
230     \csundef{\@nameauth@CleanName{#1#2!NF!}}}%
231     {\IfValueTF{#3}%
232     {\csundef{\@nameauth@CleanName{#2#3!MN!}}}%
233     \csundef{\@nameauth@CleanName{#2#3!NF!}}}%
234     {\csundef{\@nameauth@CleanName{#2!MN!}}}%
235     \csundef{\@nameauth@CleanName{#2!NF!}}}%
236     }%
237 }
```

`\ForgetName` undefines control sequences to trigger the “first use” option of `\Name`.

`\NamesInactive`

```

238 \newcommand{\NamesInactive}{\@nameauth@DoFormatfalse}
```

This macro deactivates formatting, even as its counterpart below activates it.

`\NamesActive`

```

239 \newcommand{\NamesActive}{\@nameauth@DoFormattrue}
```

Change History

v0.7		
General: Initial version	1	
v0.75		
General: New features described; documentation clarified	1	
\ForgetName: New parameter added	21	
\IndexName: Optional parameter added; mandatory parameter deleted	21	
\Name: Added “sobriquet” feature	16	
v0.8		
\@nameauth@CheckDot: Renamed macro to help compatibility	15	
\@nameauth@CleanName: Renamed macro to help compatibility	14	
\@nameauth@EvalDot: Renamed macro to help compatibility	15	
\@nameauth@FmtName: Renamed macro to help compatibility	15	
\@nameauth@TestDot: Renamed macro to help compatibility	15	
General: Added quick start guide for the impatient	3	
Expanded description of func- tionality	12	
Refactoring to improve func- tionality and compatibility; docu- mentation expanded	1	
\Name: Merged all major func- tionality here	16	
v0.85		
\@nameauth@FmtName: Add comma suppression	15	
\@nameauth@NoComma: Add comma handling routine	14	
\@nameauth@Root: Divide suffix handling into functional parts	14	
\@nameauth@Suffix: Add suffix handling functionality	14	
\@nameauth@Truncate: Add suffix handling functionality	14	
\@nameauth@Zapsuffix: Revise ex- tant suffix handling	15	
General: Added comma suppression and additional functionality; re- vised documentation	1	
Added new class options	2	
Added nocomma, comma op- tions	13	
New suffix removal features	6	
Updated quick start guide	3	
\AKA: Add comma suppression, ltx- doc compatibility	20	
\IndexName: Add comma suppres- sion, ltxdoc compatibility	21	
\Name: Add comma suppression to indexing, ltxdoc compatibility	16	
\PName: Add comma suppression	20	
v0.86		
General: Fixed some regressions	1	
\AKA: Fixed some regressions	20	
\IndexName: Slight tweak using \edef	21	
\Name: Slight tweak using \edef	16	

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.

Symbols	... <u>27</u> , 66, 81,	G
<code>\@nameauth@CheckDot</code>	104, 119, 141, 156	Goethe, Johann Wolf-
..... <u>30</u> , 168, 201	<code>\@nameauth@dtxfalse</code> 19	gang von 5
<code>\@nameauth@CleanName</code>	<code>\@nameauth@dtxtrue</code> . 18	Gregorio, Enrico 1
. <u>20</u> , 55, 61, 70,		
76, 85, 93, 99,	A	H
108, 114, 123,	Adams, John Q. ... 6, 7	Hammerstein II, Oskar 6
130, 136, 145,	<code>\AKA</code> 10, 173, <u>175</u>	Henry VIII 6
151, 160, 184,	Antiochus IV	
185, 193, 194,	Epiphanes .. 6, 8	I
208, 214, 219,	Arouet, François-Marie	<code>\IndexName</code> 8, <u>203</u>
229, 230, 232–235 <i>see</i> Voltaire	
<code>\@nameauth@DoCommafalse</code>		J
..... 11	B	Jean sans Peur 9, 12
<code>\@nameauth@DoCommatrue</code>	Berenstain, Jan 8	Jean the Fearless ...
..... 12	Berenstain, Stan 8	<i>see</i> Jean sans Peur
<code>\@nameauth@DoFormatfalse</code>	Boethius 5	
..... 6, 238	C	K
<code>\@nameauth@DoFormattrue</code>	Carnap, Rudolph ... 13	King Jr., Martin
..... 5, 239	Charles the Bald 5	Luther 7
<code>\@nameauth@EvalDot</code> .	Chiang Kai-shek 6	L
..... 30, <u>31</u>	Cicero, M.T. 5, 7	Louis XIV 7
<code>\@nameauth@FmtName</code> .	Clemens, Samuel L. .	Lueck, Uwe 1, 15
... <u>38</u> , 56, 63,	.. <i>see</i> Twain, Mark	
66, 71, 78, 81,	Confucius 5	M
86, 94, 101, 104,		Maimonides 10
109, 116, 119,	D	Malebranche, Nicolas 13
124, 131, 138,	De Wette, Wilhelm	Mencius 6
141, 146, 153,	M.L. 7, 8	Moses ben-Maimon ..
156, 161, 183, 192	<i>Doctor Angelicus</i> . <i>see</i>	.. <i>see</i> Maimonides
<code>\@nameauth@NoComma</code> .	Thomas Aquinas	
..... <u>24</u> , 45,	Dongen, Marc van .. 1, 18	N
46, 52, 179, 180, 206		<code>\Name</code> 5, <u>49</u> , 172
<code>\@nameauth@Punctfalse</code>	E	<code>\Name*</code> 5
..... 35	Einstein, Albert 5	<code>\NamesActive</code> ... 13, <u>239</u>
<code>\@nameauth@Puncttrue</code> 34	Eliot, George 10	<code>\NamesFormat</code>
<code>\@nameauth@Root</code> ...	Evans, Mary Anne 7–10, 12, 42, 46
..... <u>21</u> , 25, 28	. <i>see</i> Eliot, George	<code>\NamesInactive</code> <u>238</u>
<code>\@nameauth@Suffix</code> <u>22</u> , 25		<code>\NamesInctive</code> 13
<code>\@nameauth@TestDot</code> .	F	O
..... <u>32</u> , 40	<code>\ForgetName</code> 13, <u>226</u>	Oberdiek, Heiko ... 1, 14
<code>\@nameauth@Truncate</code> <u>22</u>		
<code>\@nameauth@Zapsuffix</code>		

P		
\PName	9, <u>170</u>	
Ptolemy I Soter	6	
R		
Rambam	<i>see</i>	
<i>also</i> Maimonides		
Rambam	10	
S		
Snel van Royen, Rudolph	11	
T		
Thomas Aquinas	9, 12	
V		
Vlad II Dracul	12	
Vlad III Dracula	12	
Voltaire	10	
W		
Walther, C.F.W.	8	