The nameauth package

Charles P. Schaum
charles dot schaum at att dot net

v0.9 from 2012/02/10

Abstract

Using the nameauth package, an author can encode names according to a name authority. Index entries will be consistent if the input parameters are consistent. An author can move blocks of text arbitrarily and the names will be reformatted automatically, making it easier to transition from drafts to a final manuscript. This package mainly supports Western naming conventions, with some basic features for ancient, royal, and Eastern names.

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 form of people's names. You would use a name authority to index and keep track of name forms. Your index might use abbreviated forms. Either a substantive or copy editor would keep track of the main form, the variants, and their relationship to the index. That might 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, making that person more desirable for publication.

1.1 Design Decisions

This package assumes that an author or editor wants to minimize keystrokes. The default behavior consists of the most likely settings. Considerable variation from the defaults is possible. The `.dtx` file is an invaluable source of information on how many variations of the macros can be used.

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. The class options are listed with defaults first in each category:

`mainmatter`	The default behavior triggers special typesetting of the first occurrence of a name, starting at the beginning of a document.
`frontmatter`	This option suppresses the special typesetting of the first occurrence of a name before the invocation of `\NamesActive`. 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. *Note*: One can switch at will between formatted and non-formatted sections; see Section 2.8.
`smallcaps`	The default behavior when a name is first encountered is to print it in small caps.
`italic`	This option causes the first occurrence of the name to be italicized.
`boldface`	This option causes the first occurrence of the name to be set in boldface.
`noformat`	This option suppresses document formatting after the invocation of `\NamesActive`. If an author wants the indexing and aliasing functions without any special typesetting, this option accomplishes that easily. *Note*: One can override the name typesetting options manually; see Section 2.7.1. This option is *not* equivalent to `frontmatter`. The latter functions independently and never produces any formatting. This approach allows one to toggle formatting on and off without retyping the document. With `\NamesFormat` It anticipates variation among publishers' house styles.
`nocomma`	The default behavior suppresses printing of commas between surnames and suffixes, following modern styles like *Chicago Manual of Style*. See Section 2.5.4 for implications of this behavior.
`comma`	Print commas between surnames and suffixes, following older styles. See also Section 2.5.4.

2.2 Example Text

We begin with a sample text that attempts to cover the basic features, and even a couple of advanced ones. It is in the genre of a merry wedding reception toast, proving that a serious package can be used lightheartedly.

DISCLAIMER: I use a number of names associated with historical figures throughout this document. This is because I expect that the users of this package will refer to real-world figures. At no time in this document am I intending either to promote, disparage, align with, align against, or make any assertions about any persons living or dead. As far as I am concerned, all names mentioned herein deserve respect for the impact and legacy of their bearers.

Example Text:

```
This is a toast to \Name[John]{Smith} and his childhood sweetheart,
\Name[J.Q.]{Public}[Jane Q.]. \FName[John]{Smith} and
``\FName[J.Q.]{Public}[Janie]'' have finally made it official today!
They first met in college, where they had to learn about
\Name{Aristotle} and all the stuff that I couldn't keep in my head.
[laughs] Nevertheless, by the time they got to
\Name{John}[Duns Scotus] they were definitely a number. They studied
medieval history-makers like \Name{Gregory, I}
``\AKA*{Gregory, I}{Gregory}[the Great]'' and by the Renaissance they
were engaged. After spending time in separate grad schools, these
promising medievalists got faculty positions at adjoining colleges and
here we are. As the brother of \FName[J.Q.]{Public}[Jane]
``\AKA*[J.Q.]{Public}{Jane}[the Great]'' I am happy to welcome
\FName[John]{Smith}[Sir John] to our raucous family.
```

This is a toast to John Smith and his childhood sweetheart, Jane Q. Public. John and "Janie" have finally made it official today! They first met in college, where they had to learn about Aristotle and all the stuff that I couldn't keep in my head. [laughs] Nevertheless, by the time they got to John Duns Scotus they were definitely a number. They studied medieval history-makers like Gregory I "the Great" and by the Renaissance they were engaged. After spending time in separate grad schools, these promising medievalists got faculty positions at adjoining colleges and here we are. As the brother of Jane "the Great" I am happy to welcome Sir John to our raucous family.

Here you will notice that one can make a reference to Pope `\Name{Gregory, I}` followed by "the Great" via ``` ``\AKA*{Gregory, I}{Gregory}[the Great]'' ```. That is achieved by the ambiguous behavior of the default `nocomma` option plus the "sobriquet only" feature of `\AKA*`. It is easy to use these features, but can be tricky to debug. Sections 2.5.4 and 2.5.5 have more details.

2.3 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 help you get used to the package, but they do not show all possibilities.

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.
4. Checking index references will help you find mistakes.
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 in the text)	`\Name*[John]{Smith}` or `\Name[John]{Smith}` or `\FName[John]{Smith}`
Full Western name	`\Name*[John]{Smith}`
Short surname only (later)	`\Name[John]{Smith}`
First name only (later)	`\FName[John]{Smith}`
Print name with alternate forenames/nicknames in the text, not in the index (first reference)	`\Name*[J.Q.]{Public}[Jane Q.]` or `\Name[J.Q.]{Public}[Jane Q.]` or `\FName[J.Q.]{Public}[Jane Q.]`
Nickname only (later)	`\FName[J.Q.]{Public}[Jane Q.]`
Full name without longer form	`\Name*[J.Q.]{Public}`
Same person, last name only (subsequent reference)	`\Name[J.Q.]{Public}[Jane Q.]` or †`\Name[J.Q.]{Public}`
Print an ancient name	`\Name{Plato}` or `\Name*{Plato}`
Print a full Eastern name (first reference)	`\Name*{Mao}[Tse-tung]` or `\Name{Mao}[Tse-tung]`
Full Eastern name	`\Name*{Mao}[Tse-tung]`
Short name (later reference)	`\Name{Mao}[Tse-tung]`
Print a full "royal" name (first reference)	`\Name*{Louis}[XIV]` or `\Name{Louis}[XIV]`
Full "royal" name	`\Name*{Louis}[XIV]`
Shorter subsequent reference	`\Name{Louis}[XIV]`
Ancient name and sobriquet	`\Name*{Antiochus V}[Eupator]` or `\Name{Antiochus V}[Eupator]`
Shorter subsequent reference	`\Name{Antiochus V}[Eupator]`

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

I strongly suggest reading Section 2.5.4 before using comma-delimited suffixes.

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

† 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 name that is only a *see* reference to another.	`\AKA[Bob]{Hope}[Leslie Townes]{Hope}`
Print both names, the latter in parens	`\PName[Bob]{Hope}[Leslie Townes]{Hope}`
Print an ancient name, number, and sobriquet	§`\AKA*{Gregory, I}{Gregory}[the Great]`
Mononym followed by a lesser-known name	`\PName{Prince}[Prince Rogers]{Nelson}`
Flexibly associate a name with a lesser-known name	¶... a tribute to `\Name{Ari Up}`, born `\AKA{Ari Up}[Arianne]{Forster}`...

§ This depends on the default `nocomma` option.

¶ `{Ari Up}` is one unit; one would not index the stage name as "Up, Ari".

Avoiding the following pitfalls will save time and frustration:

1. Mixing sobriquets with modern forms fails due to the "alternate 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 Section 2.5.4 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 in macro arguments will create incorrectly sorted index entries. Take care to avoid them.

2.4 Basic Macros: \Name and \FName

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

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:

FNN	*SNN*	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 respective printed and indexed forms, usually *FNN SNN* and *SNN*, *FNN*. This takes care of most Western names. For those with one name, such as ancient figures or stage names, one can drop the *FNN* so that `\Name` produces the result *SNN* for both text and index.`\Name` always prints the surname or "base name."

Sometimes you might want to have the option of using either an alternate set of forenames, like a nickname, 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 may choose to include or exclude nicknames and such, but you must always use the sobriquet form of a name consistently.

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 more examples:

`\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: I am playing a "dirty trick" by making each line begin as if the name had not yet occurred. This trick can be used, for example, to force the formatting of the first name in a chapter or section. See Section 2.9 for more.

2.4.2 Tabular Summary of `\Name`

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

FNN	*SNN*	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

Here are examples that show variation among alternate names. As long as the "main" *FNN* are constant, so will the index entries be constant:

In all cases, the index entries contain "De Wette, Wilhelm M.L."

`\Name[Wilhelm M.L.]{De Wette}[Wilhelm Martin Leberecht]`
Wilhelm Martin Leberecht De Wette (text, first occurrence)
De Wette (text, second occurrence)

`\Name*[Wilhelm M.L.]{De Wette}`
Wilhelm M.L. De Wette (text, first occurrence)
Wilhelm M.L. De Wette (text, second occurrence)

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.

Another option employs a "sobriquet" feature for royal names and basic Eastern names. `\AKA` and `\PName` cannot refer to these forms, although they can use these forms in the *second* name argument to construct a cross-reference. A workaround is discussed in Section 2.5.4. The following method is the only one that works with the `comma` option. Valid "sobriquet" forms are:

FNN	*SNN*	Alternate Names	Result	Function
(none)	`Henry`	`VIII`	Henry VIII	`\Name*{Henry}[VIII]`
(none)	`Henry`	`VIII`	Henry	`\Name{Henry}[VIII]`
(none)	`Chiang`	`Kai-shek`	Chiang Kai-shek	`\Name{Chiang}[Kai-shek]`
(none)	`Chiang`	`Kai-shek`	Chiang	`\Name{Chiang}[Kai-shek]`

2.4.3 \FName

\FName This casual friend of \Name prints only "first" names except if a first use occurs, whereupon it prints a full, formatted name as set by the class options or the formatting macros. The syntax is basically the same:

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

Remember that \FName *has no starred form.* Next we see what it does:

FNN	*SNN*	Alternate Names	Result
Albert	Einstein	(none)	Albert
(none)	Confucius	(none)	Confucius
M.T.	Cicero	Marcus Tullius	Marcus Tullius
(none)	Charles	the Bald	Charles

If one accidentally referred to \FName[Max]{Planck} as a first reference, that would appear as Max Planck. otherwise it would just be Max. Using the "alternate names" option prints them instead. For example, aviation hero Chesley B. Sullenberger III can be noted as:

``\FName[Chesley B.]{Sullenberger, III}[Sully]'' "Sully"

A good way to cut keystrokes would be to assign the above macro to the control sequence \Sully. With comma-delimited suffixes we note special cases governed by the nocomma class option (see Section 2.5.4). These include names like like J.D. Rock III and Charles V:

\FName[J.D.]{Rock, III}, "J.D.," \FName{Charles, V}, "Charles"

Mao Tse-tung, "Mao," does not work with \FName because it is an Eastern name. \FName works best with Western names and mononyms.

\FName understands that an extra period could result from typing:

```
My boss is \Name*[J.D.]{Rock, III}.
We all call him \FName[J.D.]{Rock, III}.
```

\FName solves this by scanning for and eliminating that extra period:

My boss is J.D. Rock III.
We all call him J.D.

2.5 Advanced Topics

2.5.1 Hyphenation

I find it helpful to use the babel or polyglossia packages to help with name hyphenation. If one is using English as the main language, the default hyphenation patterns may not suffice. For example, the name John Strietelmeier

may break thus: "Stri-etelmeier." That is fixed by creating a `\de` macro equivalent to `\newcommand{\de}[1]{\foreignlanguage{ngerman}{#1}}` (using babel) and writing `\de{\Name[John]{Strietelmeier}}`.

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

2.5.2 `\IndexName`

`\IndexName` This macro allows one to create an index entry with the same format as `\Name` and friends. It prints no text in the body and can be used like `\index`, but without any special formatting. The syntax is similar to `\Name`:

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

`\IndexName` does not index the alternate names unless no *FNN* are present, at which time is acts like `\Name` using the sobriquet feature.

2.5.3 Manual Index Entries

Starting with version 0.9, `\Name` and friends produce index entries that are compatible with manual index entries. The comma and suffix handling are now entirely expandable. A great way to see how these index entries are formed is to run latex and friends using makeindex—I have not tested this package with `xindy`—and examine the `.ind` file.

2.5.4 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 input 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 *SNN* argument of `\Name` and friends will cause unwanted results. Fortunately, that is unlikely.

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

`\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.

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:

FNN	*SNN*	Alternate Names	Result	Function
(none)	`Louis, XIV`	(none)	Louis XIV	`\Name{Louis, XIV}`
(none)	`Louis, XIV`	(none)	Louis	`\Name{Louis, XIV}`
(none)	`Sun, Yat-sen`	(none)	Sun Yat-sen	`\Name{Sun, Yat-sen}`
(none)	`Sun, Yat-sen`	(none)	Sun	`\Name{Sun, Yat-sen}`

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

Even though suffix and sobriquet features look like they produce the same *output* in the body text using the `nocomma` option, they are internally *different*. They will not respect each other regarding "first use," although they will (tentatively) cooperate in the index. Use each approach consistently. An example of "dangerous" use of these features occurs in Section 2.7. The `comma` option will cause these forms above to have commas and behave differently. Again, *caveat auctor*.

2.5.5 Pen Names: An Introduction

The macro `\AKA` deals with pseudonyms, stage names, *noms de plume*, etc. We already saw the strengths and weaknesses of the suffix feature above as a workaround for sobriquets and Eastern names when using `\AKA`. Before we examine its function in detail, we touch on the only solution for some name forms, especially if one chooses the `comma` option:

```
\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.

The suffix workaround also can work with this example, but not with something like `\Name{Vlad, III}[Dracula]`. In order to make that form work, you would have to use `\Name{Vlad, III Dracula}`. The latter two forms both shorten to Vlad and look the same using the `nocomma` option. If you use both forms (see Section 2.7) errors may bite. 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.

2.5.6 \AKA

\AKA
\AKA*

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

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

Notice that the *FNN* and *SNN* arguments *do not accept* the third argument field of \Name. This means that one cannot use \AKA to create a *see* reference to a name with a sobriquet or to an Eastern name unless one uses the suffix feature mentioned previously. Here is a quick review of what works and what fails:

FNN	*SNN*	Alt. *FNN*	Alt. *SNN*	Alt. names	Result
Bob	Hope	Leslie Townes	Hope	(none)	success
†Bob	Hope	Leslie Townes	Hope	Lester T.	success
(none)	Louis	XIV	Sun King	(none)	FAIL
‡(none)	Louis, XIV	(none)	Sun King	(none)	success
(none)	Gregory	I	Gregory	the Great	FAIL
§(none)	Gregory, I	(none)	Gregory	the Great	success

† This succeeds, but replaces "Leslie Townes" with "Lester T."
‡ This form uses the nocomma feature.
§ This produces different output, depending on whether \AKA or \AKA* is used.

Since \AKA is designed to handle a number of otherwise incompatible needs, its use may be the most complex of all macros in this package. \AKA creates a cross reference. The target of this cross-reference is either *SNN*, *FNN* or just *SNN*. The main entry in which the cross-reference occurs is constructed exactly like \Name handles its arguments. The *alternate FNN* are replaced by the *alternate names* if both exist. The *alternate names* follow the *alternate SNN* otherwise.

The twist is the starred form \AKA*. If the starred form is used with the template (none){⟨*alternate SNN*⟩}[⟨*alt. names*⟩], it only prints the [⟨*alt. names*⟩]. This allows it to be used in the manner of the example text in Section 2.2. \AKA prints {⟨*alternate SNN*⟩} followed by [⟨*alt. names*⟩]. A difference between the behavior of \AKA* and \Name using sobriquets is that \AKA* creates a cross-reference and allows the sobriquet to be printed separately from the main name within some manner of punctuation.

\AKA only prints the alternate name. It assumes that a \Name macro occurs somewhere to create the page-indexed target of a cross-reference. No error checking otherwise occurs for this. The macro also prevents double periods.

A brief example 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.

The alternate name references generated by `\AKA` and `\AKA*` only work as cross-references. If you include these alternate names in the parameters of `\Name`, it will print the alternate names in the text and emit a warning. It will not index the alternate names at that point.

In some cases, that is enough. In other cases, one might wish to index an alternate name with page numbers. In that case, do not use `\AKA`. Use `\Name` for both the main name and the alternate name. You will have to create manual cross-references with `\index`. For example:

Authoritative Name	Alternate Name	Example of Use
Maimonides	Moses ben-Maimon	`\AKA{Maimonides}{Moses ben-Maimon}`
Maimonides	Rambam	`\Name{Rambam}%`
		`\index{Rambam|seealso{Maimonides}}`

`\AKA` will not create multiple instances of a cross-reference. This prevents possible errors 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). One must add a manual index entry:

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

2.5.7 `\PName`

`\PName` `\PName` is a "convenience macro" that sacrifices some of the flexibility of `\AKA` for simplicity. It does not implement `\AKA*` and it works best with Western-style names. Like `\Name`, `\PName` prints a Western-style "main" name using either starred or unstarred forms that affect only that name. Like `\AKA`, it follows with an alternate name, but prints the full alternate name in parentheses. `\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}
\PName[Mark]{Twain}[Samuel L.]{Clemens}
```
Print Mark Twain (Samuel L. Clemens) the first time it appears. Later, print Mark Twain (Samuel L. Clemens). The form `\PName` later just prints Twain (Samuel L. Clemens).

```
\PName*{Voltaire}[François-Marie]{Arouet}
\PName{Voltaire}[François-Marie]{Arouet}
```
Print Voltaire (François-Marie Arouet) the first time it appears. Later, print Voltaire (François-Marie Arouet). Both forms do the same thing in this case.

If you use the unstarred forms `\PName` and `\Name`, you must remember that, just because you include a full name as a parameter does not mean that the full name will print. You could end up with ambiguous references to the same last name, e.g., "Snel van Royen." You are responsible for checking this.

2.6 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
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

Other accents will not work unless you use TEX control sequences or XƎLATEX. 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{ſ}{\textlongs}
\newunicodechar{ā}{\=a}
\newunicodechar{ḿ}{\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.

2.7 Name Formatting

2.7.1 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 Ţepeş, "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 became 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.

After using `\let` to revert `\NamesFormat`, a first occurrence again takes the form: Vlad III Dracula, while subsequent references are to Vlad III.

Let me again stress that "royal" names used in this manner do not work with `\PName` and `\AKA`. Consistently use either the suffix mechanism (Section 2.5.4) or use as a guide the examples above for Jean sans Peur and Thomas Aquinas. If you use the suffix mechanism, you would use the forms:

```
\Name{Vlad III, Dracula} became 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 became known as VladŢepeş, "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.

NOTE: The "Dracula" example is complex and the usual "first use" feature has been manipulated extensively to produce these results. If you mix casually the sobriquet and suffix forms, errors may "bite." Vlad III was a complex historical figure, as was his father, living in the no-man's land between the Ottoman Empire and the Holy Roman Empire. The story of the Jagiellon monarchy of Hungary,

later to come under Hapsburg control, is also complex, as was that of Ivan IV "the Terrible" in fifteenth-century Muscovy. I am not trying to saddle a historical figure with the fiction of Bram Stoker, although the latter is probably responsible for much tourism regarding Vlad III.

2.8 Formatting Certain Sections

`\NamesActive` `\NamesInctive`

Using the `frontmatter` option deactivates formatting until `\NamesActive` occurs. 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`:

`\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

2.9 Tweaks: `\ForgetName` and `\SubvertName`

Using these two macros may hinder the arbitrary rearrangement of text. I suggest that one wait until the final draft before implementing them. They either force or suppress the active formatting of names.

`\ForgetName`

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

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

This macro causes `\Name` and friends to "forget" prior uses of a name with respect to typesetting. The next use will print as if it were a "first use." Index entries and pseudonyms (see above) are *never* forgotten.

`\SubvertName`

This macro is the opposite of the one above. It takes the same parameters. It handles its arguments in the same manner as `\Name`, except that it ignores the final parameter if *FNN* are present. The syntax is:

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

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

3 Implementation

3.1 Class Options and Required Packages

```
\newif\if@nameauth@DoFormat
\newif\if@nameauth@DoComma
```

These Boolean values are used to control formatting and comma suppression. Both may be set with class options, but only `@nameauth@DoFormat` can be toggled with use interface macros `\NamesActive` and `\NamesInactive`.

```
\newif\if@nameauth@Comma
\newif\if@nameauth@Punct
```

These Boolean values are used internally for detection of suffixes and final periods.

```
\DeclareOption{mainmatter}{\@nameauth@DoFormattrue}
\DeclareOption{frontmatter}{\@nameauth@DoFormatfalse}
\DeclareOption{smallcaps}{\newcommand{\NamesFormat}{\scshape}}
\DeclareOption{italic}{\renewcommand{\NamesFormat}{\itshape}}
\DeclareOption{boldface}{\renewcommand{\NamesFormat}{\bfseries}}
\DeclareOption{noformat}{\renewcommand{\NamesFormat}{}}
\DeclareOption{nocomma}{\@nameauth@DoCommafalse}
\DeclareOption{comma}{\@nameauth@DoCommatrue}
\ExecuteOptions{smallcaps,mainmatter,nocomma}
\ProcessOptions\relax
\RequirePackage{etoolbox}
\RequirePackage{xparse}
```

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. The other options can be changed without problem and several user interface macros provide that functionality.

3.2 Internal Macros

`\@nameauth@CleanName`

```
\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 parse a "base" name into a radix and a suffix. They are designed so that their function occurs completely at the time of macro expansion, not execution. This expandability is key to the proper function of this package. They form the kernel of the suffix removal and comma suppression features.

\@nameauth@Root

```
\newcommand*{\@nameauth@Root}[1]{%
    \@nameauth@TrimRoot#1\relax%
}
```

Anything starting with a comma and ending with the end of the name is stripped off. That includes "Sr.," "Jr.," "III," and so on. An extra comma is included at the end of the parameter when `\@nameauth@Root` is called directly from `\Name` and friends so that the delimiter list will always be correct. By using the comma-checking routines below, however, one can call this macro only when the parameter takes the form <a,b> and properly select the root and suffix (see below).

\@nameauth@TrimRoot

```
\def\@nameauth@TrimRoot#1,#2\relax{#1}
```

This delimited-parameter macro strips off the first parameter.

\@nameauth@Suffix

```
\newcommand*{\@nameauth@Suffix}[1]{%
    \@nameauth@TrimSuffix#1\relax%
}
```

Anything before a comma is stripped off by `\@nameauth@Suffix`, but it should be called only in a conditional governed by `@nameauth@Comma`. This macro calls its auxiliary macro below.

\@nameauth@TrimSuffix

```
\def\@nameauth@TrimSuffix#1,#2\relax{#2}
```

This delimited-parameter macro strips off the second parameter.

\@nameauth@CheckComma

```
\newcommand*{\@nameauth@CheckComma}[1]{%
    \@nameauth@CheckSuffix#1,\relax%
}
```

This macro checks for a comma-delimited suffix. It calls its auxiliary macro below.

\@nameauth@CheckSuffix

```
\def\@nameauth@CheckSuffix#1,#2\relax{%
    \def\Test{#2}%
    \ifx\Test\@empty\@nameauth@Commafalse\else\@nameauth@Commatrue\fi%
}
```

This macro checks for a comma-delimited suffix and sets the Boolean `@nameauth@Comma` accordingly.

\@nameauth@NoComma

```
\newcommand*{\@nameauth@NoComma}[1]{%
     \@nameauth@Root{#1}\@nameauth@Suffix{#1}%
}
```

This macro removes a comma from a name by breaking a <root, suffix> pair into a <root><suffix> pair. It preserves the leading space or lack thereof in the suffix.

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

\@nameauth@CheckDot

```
\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

```
\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`, `\FName`, and `\AKA`.

\@nameauth@TestDot

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

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

```
\DeclareDocumentCommand\@nameauth@FmtName{s m}%
{%
     \@nameauth@TestDot{#2}%
     \IfBooleanTF{#1}%
          {#2}%
          {\bgroup\NamesFormat{#2}\egroup}%
}
```

`\@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. The main reason for making this a separate macro was to offer a means of adding features in modular fashion.

3.3 User Interface Macros

`\Name`

```
\DeclareDocumentCommand\Name{s o m o}%
{%
     \def\Surnames{#3}%
     \if@nameauth@DoComma\else%
          \@nameauth@CheckComma{#3}%
          \if@nameauth@Comma%
               \edef\Surnames{\@nameauth@Root{#3}\@nameauth@Suffix{#3}}%
          \fi%
     \fi%
     \IfValueTF{#2}%
     {\IfValueTF{#4}{\edef\Forenames{#4}}{\edef\Forenames{#2}}%
     \ifcsname\@nameauth@CleanName{#2#3!PN!}\endcsname%
          \expandafter\@nameauth@FmtName\expandafter*\expandafter{%
               \expandafter\Forenames\expandafter\space\Surnames}%
          \PackageWarning{nameauth}%
          {You cannot create a page reference from the xref: #2 #3.}%
     \else%
          \if@nameauth@DoFormat%
               \ifcsname\@nameauth@CleanName{#2#3!MN!}\endcsname%
               \IfBooleanTF{#1}%
                    {\expandafter\@nameauth@FmtName\expandafter*\expandafter{%
                         \expandafter\Forenames\expandafter\space\Surnames}%
                         \index{\Surnames, #2}}%
                    {\@nameauth@FmtName*{\@nameauth@Root{#3,}}%
                         \index{\Surnames, #2}}%
               \else%
                    \csgdef{\@nameauth@CleanName{#2#3!MN!}}{}%
                    \expandafter\@nameauth@FmtName\expandafter{%
                         \expandafter\Forenames\expandafter\space\Surnames}%
                         \index{\Surnames, #2}%
               \fi%
          \else%
               \ifcsname\@nameauth@CleanName{#2#3!NF!}\endcsname%
               \IfBooleanTF{#1}%
                    {\expandafter\@nameauth@FmtName\expandafter*\expandafter{%
                         \expandafter\Forenames\expandafter\space\Surnames}%
                         \index{\Surnames, #2}}%
                    {\@nameauth@FmtName*{\@nameauth@Root{#3,}}%
                         \index{\Surnames, #2}}%
               \else%
```

```
                \csgdef{\@nameauth@CleanName{#2#3!NF!}}{}%
                \expandafter\@nameauth@FmtName\expandafter*\expandafter{%
                    \expandafter\Forenames\expandafter\space\Surnames}%
                    \index{\Surnames, #2}%
            \fi%
        \fi%
    \fi}%
    {\IfValueTF{#4}
        {\ifcsname\@nameauth@CleanName{#3#4!PN!}\endcsname%
            \expandafter\@nameauth@FmtName\expandafter*%
                \expandafter{\Surnames\space#4}%
            \PackageWarning{nameauth}%
            {You cannot create a page reference from the xref: #3 #4.}%
        \else%
            \if@nameauth@DoFormat%
                \ifcsname\@nameauth@CleanName{#3#4!MN!}\endcsname%
                \IfBooleanTF{#1}%
                    {\expandafter\@nameauth@FmtName\expandafter*%
                        \expandafter{\Surnames\space#4}%
                        \index{\Surnames\space#4}}%
                    {\@nameauth@FmtName*{\@nameauth@Root{#3,}}%
                        \index{\Surnames\space#4}}%
                \else%
                    \csgdef{\@nameauth@CleanName{#3#4!MN!}}{}%
                    \expandafter\@nameauth@FmtName\expandafter{%
                        \Surnames\space#4}%
                        \index{\Surnames\space#4}%
                 \fi%
            \else%
                \ifcsname\@nameauth@CleanName{#3#4!NF!}\endcsname%
                \IfBooleanTF{#1}%
                    {\expandafter\@nameauth@FmtName\expandafter*%
                        \expandafter{\Surnames\space#4}%
                        \index{\Surnames\space#4}}%
                    {\@nameauth@FmtName*{\@nameauth@Root{#3,}}%
                        \index{\Surnames\space#4}}%
                \else%
                    \csgdef{\@nameauth@CleanName{#3#4!NF!}}{}%
                    \expandafter\@nameauth@FmtName\expandafter*%
                        \expandafter{\Surnames\space#4}%
                        \index{\Surnames\space#4}%
                 \fi%
            \fi%
        \fi}%
        {\ifcsname\@nameauth@CleanName{#3!PN!}\endcsname%
            \expandafter\@nameauth@FmtName\expandafter*%
                \expandafter{\Surnames}%
            \PackageWarning{nameauth}%
            {You cannot create a page reference from the xref: #3.}%
        \else%
```

```
                \if@nameauth@DoFormat%
                     \ifcsname\@nameauth@CleanName{#3!MN!}\endcsname%
                     \IfBooleanTF{#1}%
                          {\expandafter\@nameauth@FmtName\expandafter*%
                               \expandafter{\Surnames}%
                               \index{\Surnames}}%
                          {\@nameauth@FmtName*{\@nameauth@Root{#3,}}%
                               \index{\Surnames}}%
                     \else%
                          \csgdef{\@nameauth@CleanName{#3!MN!}}{}%
                          \expandafter\@nameauth@FmtName\expandafter{\Surnames}%
                               \index{\Surnames}%
                      \fi%
                \else%
                     \ifcsname\@nameauth@CleanName{#3!NF!}\endcsname%
                     \IfBooleanTF{#1}%
                          {\expandafter\@nameauth@FmtName\expandafter*%
                               \expandafter{\Surnames}%
                               \index{\Surnames}}%
                          {\@nameauth@FmtName*{\@nameauth@Root{#3,}}%
                               \index{\Surnames}}%
                       \else%
                          \csgdef{\@nameauth@CleanName{#3!NF!}}{}%
                          \expandafter\@nameauth@FmtName\expandafter*%
                               \expandafter{\Surnames}%
                               \index{\Surnames}%
                     \fi%
                \fi%
           \fi}%
      }%
      \if@nameauth@Punct\expandafter\@nameauth@CheckDot\fi%
}
```

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

The reason why one sees seven outcomes repeated three times for twenty-one variations is because one must make different decisions based on the dynamic interaction of many factors beyond mere input. Three choices occur when one tries to use an "alternate name" defined by `\AKA` as an argument to `\Name`. The latter prints the arguments with comma suppression if appropriate, emits a warning, and exits. One cannot use `\ForgetName` to expunge a pen name. This is a deliberate decision to avoid corruption of the index cross-references.

Regarding the remaining eighteen 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 Boolean and value parameters offer eight valid patterns of input. The first or subsequent use are two more, as are front or main matter use. Operations on suffixes are governed by the first or subsequent use. The formatting or non-formatting of output is governed both by a formatting Boolean and by first/subsequent use. A final choice deals with handling alternate names in the `\Forenames` macro.

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` and `\@nameauth@FmtName`.

`\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.

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. Note again that one must always use commas with suffixes in the input, even if commas are suppressed in the output.

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.

`\FName`

```
\DeclareDocumentCommand\FName{o m o}%
{%
     \def\Surnames{#2}%
     \if@nameauth@DoComma\else%
          \@nameauth@CheckComma{#2}%
          \if@nameauth@Comma%
               \edef\Surnames{\@nameauth@Root{#2}\@nameauth@Suffix{#2}}%
          \fi%
     \fi%
```

```
\IfValueTF{#1}%
 {\IfValueTF{#3}{\edef\Forenames{#3}}{\edef\Forenames{#1}}%
 \ifcsname\@nameauth@CleanName{#1#2!PN!}\endcsname%
      \expandafter\@nameauth@FmtName\expandafter*%
           \expandafter{\Forenames}%
      \PackageWarning{nameauth}%
      {You cannot create a page reference from the xref: #1 #2.}%
 \else%
      \if@nameauth@DoFormat%
           \ifcsname\@nameauth@CleanName{#1#2!MN!}\endcsname%
                \expandafter\@nameauth@FmtName\expandafter*%
                     \expandafter{\Forenames}%
                \index{\Surnames, #1}%
           \else%
                \csgdef{\@nameauth@CleanName{#1#2!MN!}}{}%
                \expandafter\@nameauth@FmtName\expandafter{%
                    \expandafter\Forenames\expandafter\space\Surnames}%
                \index{\Surnames, #1}%
           \fi%
      \else%
           \ifcsname\@nameauth@CleanName{#1#2!NF!}\endcsname%
               \expandafter\@nameauth@FmtName\expandafter*%
                     \expandafter{\Forenames}%
               \index{\Surnames, #1}%
           \else%
                \csgdef{\@nameauth@CleanName{#1#2!NF!}}{}%
                \expandafter\@nameauth@FmtName\expandafter*\expandafter{%
                     \Forenames\expandafter\space\Surnames}%
                \index{\Surnames, #1}%
           \fi%
      \fi%
 \fi}%
 {\IfValueTF{#3}
      {\ifcsname\@nameauth@CleanName{#2#3!PN!}\endcsname%
           \expandafter\@nameauth@FmtName\expandafter*%
                \expandafter{\Surnames\space#3}%
           \PackageWarning{nameauth}%
           {You cannot create a page reference from the xref: #2 #3.}%
      \else%
           \if@nameauth@DoFormat%
                \ifcsname\@nameauth@CleanName{#2#3!MN!}\endcsname%
                     \@nameauth@FmtName*{\@nameauth@Root{#2,}}%
                     \index{\Surnames\space#3}%
                \else%
                     \csgdef{\@nameauth@CleanName{#2#3!MN!}}{}%
                     \expandafter\@nameauth@FmtName\expandafter{\Surnames\space#3}%
                     \index{\Surnames\space#3}%
                 \fi%
           \else%
                \ifcsname\@nameauth@CleanName{#2#3!NF!}\endcsname%
```

```
                        \@nameauth@FmtName*{\@nameauth@Root{#2,}}%
                        \index{\Surnames\space#3}%
                   \else%
                        \csgdef{\@nameauth@CleanName{#2#3!NF!}}{}%
                             \expandafter\@nameauth@FmtName\expandafter*%
                        \expandafter{\Surnames\space#3}%
                        \index{\Surnames\space#3}%
                    \fi%
              \fi%
         \fi}%
         {\ifcsname\@nameauth@CleanName{#2!PN!}\endcsname%
              \expandafter\@nameauth@FmtName\expandafter*%
                   \expandafter{\Surnames}%
              \PackageWarning{nameauth}%
              {You cannot create a page reference from the xref: #2.}%
         \else%
              \if@nameauth@DoFormat%
                   \ifcsname\@nameauth@CleanName{#2!MN!}\endcsname%
                        \@nameauth@FmtName*{\@nameauth@Root{#2,}}%
                        \index{\Surnames}%
                   \else%
                        \csgdef{\@nameauth@CleanName{#2!MN!}}{}%
                        \expandafter\@nameauth@FmtName\expandafter{\Surnames}%
                        \index{\Surnames}%
                    \fi%
              \else%
                   \ifcsname\@nameauth@CleanName{#2!NF!}\endcsname%
                        \@nameauth@FmtName*{\@nameauth@Root{#2,}}%
                        \index{\Surnames}%
                    \else%
                        \csgdef{\@nameauth@CleanName{#2!NF!}}{}%
                        \expandafter\@nameauth@FmtName\expandafter*%
                             \expandafter{\Surnames}%
                        \index{\Surnames}%
                   \fi%
              \fi%
         \fi}%
    }%
    \if@nameauth@Punct\expandafter\@nameauth@CheckDot\fi%
}
```

\FName is entirely derived from \Name, but scaled back to print forename(s) in a subsequent use. It obeys the nocomma behavior and eliminates double periods.

\AKA

```
\DeclareDocumentCommand\AKA{s o m o m o}%
{%
    \def\Surnamesi{#3}%
    \def\Surnamesii{#5}%
    \if@nameauth@DoComma\else%
```

```
          \@nameauth@CheckComma{#3}%
          \if@nameauth@Comma%
               \edef\Surnamesi{\@nameauth@Root{#3}\@nameauth@Suffix{#3}}%
          \fi%
          \@nameauth@CheckComma{#5}%
          \if@nameauth@Comma%
               \edef\Surnamesii{\@nameauth@Root{#5}\@nameauth@Suffix{#5}}%
          \fi%
     \fi%
     \IfValueTF{#4}%
     {\IfValueTF{#6}{\edef\Forenames{#6}}{\edef\Forenames{#4}}%
          \expandafter\@nameauth@FmtName\expandafter*\expandafter{%
               \expandafter\Forenames\expandafter\space\Surnamesii}%
          \ifcsname\@nameauth@CleanName{#4#5!PN!}\endcsname\relax\else%
               \csgdef{\@nameauth@CleanName{#4#5!PN!}}{}%
               \IfValueTF{#2}%
                    {\index{\Surnamesii, #4|see{\Surnamesi, #2}}}%
                    {\index{\Surnamesii, #4|see{\Surnamesi}}}%
          \fi}%
     {\IfValueTF{#6}%
          {\IfBooleanTF{#1}%
               {\@nameauth@FmtName*{#6}}%
               {\expandafter\@nameauth@FmtName\expandafter*\expandafter{\Surnamesii #6}}%
          \ifcsname\@nameauth@CleanName{#5#6!PN!}\endcsname\relax\else%
               \csgdef{\@nameauth@CleanName{#5#6!PN!}}{}%
               \IfValueTF{#2}%
                    {\index{\Surnamesii\space#6|see{\Surnamesi, #2}}}%
                    {\index{\Surnamesii\space#6|see{\Surnamesi}}}%
          \fi}%
          {\expandafter\@nameauth@FmtName\expandafter*\expandafter{\Surnamesii}%
          \ifcsname\@nameauth@CleanName{#5!PN!}\endcsname\relax\else%
               \csgdef{\@nameauth@CleanName{#5!PN!}}{}%
               \IfValueTF{#2}%
                    {\index{\Surnamesii|see{\Surnamesi, #2}}}%
                    {\index{\Surnamesii|see{\Surnamesi}}}%
               \fi}%
     }%
     \if@nameauth@Punct\expandafter\@nameauth@CheckDot\fi%
}
```

`\AKA` prints a pseudonym and creates index cross-references. Its starred form suppresses the "main" pseudonym if a sobriquet is passed in the alternate name parameter. It prevents multiple generation of cross-references. Like `\Name` it suppresses double periods. Its choices reflect the different choices of index references, based on the arguments it receives..

`\PName`

```
\DeclareDocumentCommand\PName{s o m o m o}%
{%
     \IfBooleanTF{#1}{\Name*[#2]{#3}}{\Name[#2]{#3}}%
```

```
319      {\space}(\AKA[#2]{#3}[#4]{#5}[#6])%
320 }
```

\PName is a convenience macro whose starred and unstarred forms call the respective versions of \Name, followed only by \AKA.

\IndexName

```
321 \DeclareDocumentCommand\IndexName{o m o}%
322 {%
323      \def\Surnames{#2}%
324      \if@nameauth@DoComma\else%
325           \@nameauth@CheckComma{#2}%
326           \if@nameauth@Comma%
327                \edef\Surnames{\@nameauth@Root{#2}\@nameauth@Suffix{#2}}%
328           \fi%
329      \fi%
330      \IfValueTF{#1}%
331           {\ifcsname\@nameauth@CleanName{#1#2!PN!}\endcsname%
332                \else%
333                     \index{\Surnames, #1}%
334                \fi}%
335           {\IfValueTF{#3}
336                {\ifcsname\@nameauth@CleanName{#2#3!PN!}\endcsname%
337                \else%
338                     \index{\Surnames\space#3}%
339                \fi}%
340                {\ifcsname\@nameauth@CleanName{#2!PN!}\endcsname%
341                \else%
342                     \index{\Surnames}%
343                \fi}%
344           }%
345 }
```

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

\ForgetName

```
346 \DeclareDocumentCommand\ForgetName{o m o}%
347 {%
348      \IfValueTF{#1}%
349           {\csundef{\@nameauth@CleanName{#1#2!MN!}}%
350           \csundef{\@nameauth@CleanName{#1#2!NF!}}}%
351           {\IfValueTF{#3}%
352                {\csundef{\@nameauth@CleanName{#2#3!MN!}}%
353                \csundef{\@nameauth@CleanName{#2#3!NF!}}}%
354                {\csundef{\@nameauth@CleanName{#2!MN!}}%
355                \csundef{\@nameauth@CleanName{#2!NF!}}}%
356           }%
357 }
```

`\ForgetName` undefines control sequences to force the "first use" option of `\Name`.

`\SubvertName`

```
\DeclareDocumentCommand\SubvertName{o m o}%
{%
     \IfValueTF{#1}%
          {\csgdef{\@nameauth@CleanName{#1#2!MN!}}%
          \csgdef{\@nameauth@CleanName{#1#2!NF!}}}%
          {\IfValueTF{#3}%
               {\csgdef{\@nameauth@CleanName{#2#3!MN!}}%
               \csgdef{\@nameauth@CleanName{#2#3!NF!}}}%
               {\csgdef{\@nameauth@CleanName{#2!MN!}}%
               \csgdef{\@nameauth@CleanName{#2!NF!}}}%
          }%
}
```

`\SubvertName` defines control sequences to suppress the "first use" of `\Name`.

`\NamesInactive`

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

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

`\NamesActive`

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

This macro is called automatically with the `mainmatter` option.

Change History

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.