The nameauth package

Charles P. Schaum
charles dot schaum at att dot net

v1.5 from 2013/02/22

Abstract

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

Contents

1 Introduction

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

- Format, print, and index names all at once.
- Automatically print a full name the first time it appears, with full or shorter forms thereafter.
- Automatically apply typesetting to names, recalling German *Sperrdruck*.
- Rearrange text without retyping the names therein.
- Allow predictable name variants in the text, yet index consistent name forms, thus abstracting the concept of a name.
- Provide for some cross-cultural features in name ordering, capitalization, and indexing.
- "Tag" and "untag" names in the index in order to index different people with the same name.
- Macros can be used in the body text, in a `\marginpar`, in tabular material, and in list environments like `enumerate`, among others.

1.1 Design Notes and Thanks

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

Thanks to Marc van Dongen, Enrico Gregorio, Philipp Stephani, Heiko Oberdiek, Uwe Lueck, and Robert Schlicht for their assistance. The light of their knowledge banished the umbra of my ignorance.

1.2 Disclaimer

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

1.3 Caveat

Throughout this manual I perform a "torture test" of sorts on several macros in this package, disregarding the advice I offer the reader. General authors may never encounter some of these cases, as in Sections 2.5.5 and 2.5.7.

2 Usage

2.1 Package Options

Package options fall into several categories. Default options are in boldface:

Show/Hide Suffix Commas

`nocomma`	**Suppress commas between surnames and suffixes, following modern styles like *Chicago Manual of Style*. Commas can be forced on a per-use basis with `\ShowComma`. See Section 2.4.1.**
`comma`	Retain commas between all surnames and suffixes. This imposes limits on certain macros.

These options are the "most global" because they affect names in both the text and the index. The `comma` option, meant for use with older texts, restricts name forms and available macros. The `\ShowComma` macro allows the general benefits of `nocomma` while allowing the more limited `comma` behavior on a per-use basis at the cost of greater complexity.

The above options are meant to be invoked at load time and left alone thereafter. Below we start those options that represent states of behavior capable of being toggled on and off.

Enable/Disable Formatting

`mainmatter`	**Enable formatting attributes (see below), starting at the beginning of a document.**
`frontmatter`	Disable formatting *before* the invocation of `\NamesActive` while retaining automatic full and short forms. This option fits well, e.g., with a foreword from a contributor. See Section 2.5.6.

Enable/Disable Indexing

`index`	**Create index entries in place with names.**
`noindex`	Prevent indexing before the invocation of the macro `\IndexActive`. See Section 2.7.5.

The options above broadly affect whether formatting and indexing are "on" or "off." They set the initial states of formatting and indexing, which can be toggled on and off respectively with `\NamesActive` and `\IndexActive` on the one hand and `\NamesInactive` and `\IndexInactive` on the other.

This package makes a distinction between *formatting*, i.e., font weight, style, family, placement, and so on, and *form*, i.e., capitalization and name order. The options on the next page alter the printing and capitalization of names apart from their formatting, but they do not change index entries:

Capitalize Entire Surnames

`normalcaps` **Do not perform any special capitalization.**

`allcaps` Capitalize entire surnames in the manner of some romanized Japanese contexts. Note that this only affects the printed form; no capitalization occurs in the index. In that case, the user should type in the caps manually.

Reverse Name Order

`notreversed` **Print names in the order specified by `\Name` and the other macros.**

`allreversed` Print name forms in "smart" reverse order. See Sections 2.4.1 and 2.4.2.

`allrevcomma` Not the same as the `comma` option or the `\ShowComma` macro. This causes all names to print in reverse Western order, with a comma between the surname and the other names.

Macros that toggle the states represented by these options include `\AllCapsActive` and `\AllCapsInactive` for capitalizing entire surnames, `\ReverseActive` and `\ReverseInactive` for reversing name order, and `\ReverseCommaActive` and `\ReverseCommaInactive` to aid making lists in last-comma-first order. An even finer grain of control exists with the macros `\CapName`, `\RevName`, and `\RevComma`, which activate these features on a per-use basis.

Formatting Attributes

`alwaysformat` With the `mainmatter` option or after `\NamesActive`, this option causes the formatting options below to affect every occurrence of `\Name`, not just the first use of a name. This recalls some contexts where German *Sperrdruck* is used. See also the material on custom formatting (Section 2.5.5).

`smallcaps` **Set the first use of a name in small caps.**

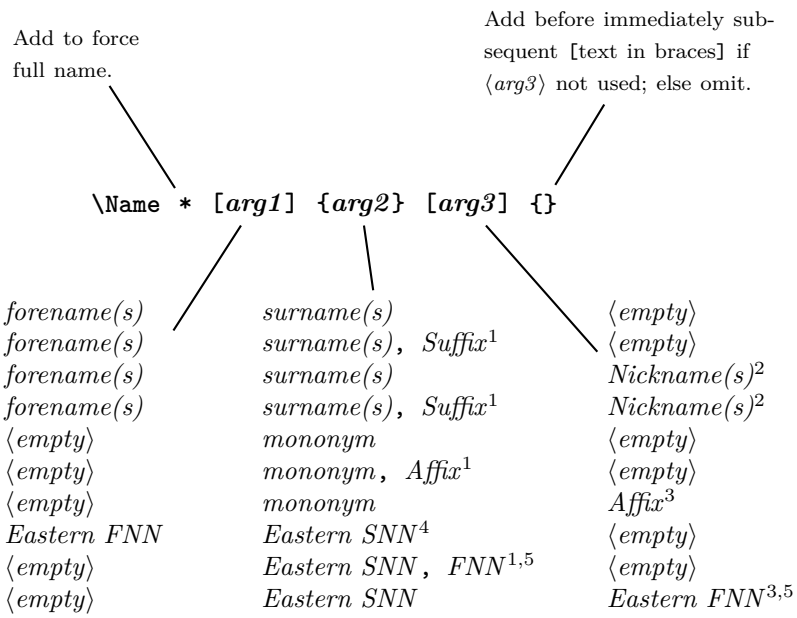
`italic` Italicize the first occurrence of a name.

`boldface` Set the first use of a name in boldface.

`noformat` This is a "do nothing" format that merely suppresses the "look" of formatting without disabling the mechanism itself.

2.2 Quick Start Guide

This general section gets one using many of the package features right away. Read the columns below in a left to right, top to bottom order. The basic patterns for `\Name[`⟨*arg1*⟩`]{`⟨*arg2*⟩`}[`⟨*arg3*⟩`]` are:



[1]The pattern `\Name[`⟨*forenames*⟩`]{`⟨*surnames, suffix*⟩`}` and all patterns thereafter with the mandatory argument ⟨*Name*$_1$*, Name*$_2$⟩ cannot be used with the `comma` option or in an instance where `\ShowComma` is used.

[2]The presence of a nickname is determined when both forename and nickname are present. In that case, the latter is swapped with the former.

[3]When ⟨*arg1*⟩ is empty, ⟨*arg3*⟩ is an affix. That form cannot be used with `\PName` and `\AKA`, but it does work with `comma` and `\ShowComma`.

[4]When expressing Eastern names via `\Name[`⟨*Eastern FNN*⟩`]{`⟨*Eastern SNN*⟩`}`, they will appear in the index as the Western form ⟨*SNN, FNN*⟩ even if `\RevName` is used to create an Eastern word order in the text.

[5]When expressing Eastern names using the form `\Name{`⟨*Eastern SNN, FNN*⟩`}` or `\Name{`⟨*Eastern SNN*⟩`}[`⟨*Eastern FNN*⟩`]`, such names will appear in the index using the Eastern form ⟨*SNN FNN*⟩ even if `\RevName` is used to create a Western word order in the text.

If given the choice, use the `nocomma` option and the ⟨*Name*$_1$*, Name*$_2$⟩ forms for suffixes and affixes in order to take advantage of extra space removal and other features that minimize possible errors.

2.2.1 Additional Hints

Q: Too many macros! I quit.

A: The macro you will use the most is `\Name`. Just use it until you discover that you need one of the other macros.

Q: I see "Paragraph ended. . . " or "Missing ⟨*grouping token*⟩ inserted" and execution stops.

A: Check that the {braces} and [brackets] are balanced.

Q: There is too much space between the initials in the names.

A: Bringhurst's *Elements of Typographic Style* calls for no spaces or thin spaces between initials. Use `\frenchspacing`.

Q: `\Name[Davey]{Jones}[a Monkee]` shows "a Monkee Jones" or just "Jones." `\FName` gives "a Monkee."

A: `\Name[`⟨*Forenames*⟩`]{`⟨*Surnames*⟩`}[`⟨*Nicknames*⟩`]` creates a nickname, not an affix or sobriquet.

Q: `\Name[Henry]{VII}` prints either "Henry VII" or "VII." Adding [Tudor] creates "Tudor VII" and "VII."

A: `\Name[`⟨*King*⟩`]{`⟨*Affix*⟩`}` will not work. Even though Mulvany, *Indexing Books*, talks about forenames and optional surnames for royals, we prevent ambiguity by encoding royal names as surnames with optional affixes. Use `\Name{`⟨*King, Affix*⟩`}` if possible, or `\Name{`⟨*King*⟩`}[`⟨*Affix*⟩`]` with the `comma` option. `\Name{Henry, VII}` gives "Henry VII" and "Henry." Manually add "Tudor" in the text and use `\TagName{Henry, VII}{, Tudor}` to add the tag automatically to the index entry.

Q: `\Name{Aethelred, the really}[Unraedig]` shows "Aethelred the really Unraedig" or "Aethelred."

A: The form `\Name{`⟨*Mononym, Affix*$_1$⟩`}[`⟨*Affix*$_2$⟩`]` creates two sobriquets, but badly. Use either `\Name{`⟨*Mononym, Affix*⟩`}` or `\Name{`⟨*Mononym*⟩`}[`⟨*Affix*⟩`]` with `comma`. Mix the forms either at your peril or at your whimsy.

Q: `\AKA{Boris}[the Animal]{Just Boris}` fails.

A: `\AKA` fails with the `{`⟨*Mononym*⟩`}[`⟨*Affix*⟩`]` pattern in order to avoid the collision of optional arguments. See Section 2.6.2.

Q: So how do I deal with some stage names and the like?

A: Use a forename or first initials to prevent failure:

OK: `\Name[J.]{Kreskin}[The Amazing]` The Amazing Kreskin (`\AKA[J.]{Kreskin}[Joseph]{Kresge}`) (Joseph Kresge)

FAIL: `\Name{Kreskin}[The Amazing]`
`\AKA{Kreskin}[Joseph]{Kresge}`

Q: How do I refer to “Iron Mike” Tyson?

A: Very carefully. One way uses ``\SubvertName[Mike]{Tyson} \FName[Mike]{Tyson}[Iron Mike]'' \Name[Mike]{Tyson} to produce the text above. Other manual methods are discussed on page 18. Using ``\AKA[Mike]{Tyson}{Iron Mike}'' creates “Iron Mike” in the text and a *see*-type cross-reference to the main name in the index.

Q: \Name has a full name in the source, but it is shorter in the text!

A: And you did not mean that to happen. Use \Name* in that case. It can be easy to forget that \Name auto-formats its arguments even if you type them out.

2.2.2 Error Handling

Version 1.5 introduces space removal in ⟨*Name*$_1$, *Name*$_2$⟩ pairs that mitigates problems in both the text and in the index. Older versions of nameauth were susceptible to fragility and moving arguments. Version 1.4 and later of nameauth address many of these issues.

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

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

Warnings result from:

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

2.3 Naming Macros

2.3.1 Surnames: \Name and \Name*

\Name
\Name*

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

\Name[⟨*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 following table helps to show how the syntax description works with first and subsequent references:

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

`\Name` connects the ⟨*FNN*⟩ to the ⟨*SNN*⟩ to create respective printed and indexed forms, as illustrated in Section 2.9 and thereafter. The invocation of `\Name` always prints the ⟨*SNN*⟩ field. `\Name` prints the "full name" at the first occurrence, then only the partial form thereafter. `\Name*` always prints the full name.

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

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

Later we shall see that a similar effect, perhaps more preferable, can be achieved with the default `nocomma` option and `\Name{Charles, the Bald}`, as seen in Section 2.4.1. One must always be consistent in using these forms. Otherwise one may get bogus index entries and errors in the text that are difficult to track.

2.3.2 Forenames: \FName

\FName
\FName*

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

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

The following table shows the output at a glance:

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

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

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

A good way to cut keystrokes would be to assign the above macro to the control sequence \Sully.

2.3.3 Full Stop Detection

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

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

See Section 2.4.1 for more on comma-delimited suffixes. Full-stop detection also works with \FName in cases like the fictional J.D. Rock III where one might discuss ``who shot \FName[J.D.]{Rock, III}.'' "who shot J.D."

2.4 Affixes and Eastern Names

2.4.1 Comma-Delimited Suffixes

Before we consider the topic of non-Western names, we must first engage the concept of suffixes and affixes as expressed in the ⟨*Name*$_1$, *Name*$_2$⟩ pattern used in the mandatory argument of `\Name`.

`\ShowComma`

The `comma` option is restrictive, but helpful in reproducing older texts. `\ShowComma` permits the default `nocomma` option, yet forces a comma to appear between the name and suffix: `\ShowComma\Name[Louis]{Gossett, Jr.}` Louis Gossett, Jr. A caveat is that one must use `\ShowComma` consistently with that name thereafter or risk errors.

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

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

We remember that these forms work because no ⟨*FNN*⟩ are present, only the `\Name{`⟨*mononym*⟩`}[`⟨*affix*⟩`]` type pattern. Of course, the mononym field can have more than one word in it, as can the affix.

Using the default package option `nocomma` with comma-delimited suffixes offers a more robust method that can handle several different name types quite flexibly. We begin with the idea of a name and a suffix. One cannot write `\Name[Oskar]{Hammerstein}[II]` without getting the likes of "II Hammerstein." On the other hand, `\Name[Oskar]{Hammerstein, II}` produces Oskar Hammerstein II the first time and Hammerstein thereafter. One must always include a comma as a suffix delimiter; trailing commas are ignored. Automatic space removal and formatting helps reduce potential error.

We can move beyond suffixes to work with more complex examples, including royal names and basic Eastern name forms:

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

The benefit to using this form is that one can type `\Name*{Louis, XIV}, the ``\AKA{Louis, XIV}{Sun King}''` and get Louis XIV, the "Sun King" in the text with an appropriate *see* reference from "Sun King" to "Louis XIV" in the index. The first method shown above prevents such usage.

Suffix and sobriquet features generally produce the same *output*, yet internally they are *different*. For example, they do not respect each other's first use. One

can force them to cooperate, as in Section 2.5.5, but it is not trivial. Users should avoid mixing the two suffix methods shown here.

2.4.2 Eastern Names

This section will work with both `\Name{`⟨*Eastern SNN*⟩`}[`⟨*Eastern FNN*⟩`]` and `\Name{`⟨*Eastern SNN, Eastern FNN*⟩`}`. We have already shown the advantages of preferring the latter form. In particular, this section addresses some features seen in romanized Japanese contexts. Nevertheless, its use may be broader.

One sees in many non-Western contexts that the family name comes first, followed by personal names. The `nameauth` package offers two routes to that result. Those routes depend on how `\Name` assembles index entries. `\Name[`⟨*Eastern FNN*⟩`]{`⟨*Eastern SNN*⟩`}` will produce an index entry of the form ⟨*Eastern SNN*⟩, ⟨*Eastern FNN*⟩. There exists a comma between the names and the default output order is Western. We will see how to change that below.

In contrast, both `\Name{`⟨*Eastern SNN*⟩`}[`⟨*Eastern FNN*⟩`]` and `\Name{`⟨*Eastern SNN, Eastern FNN*⟩`}` produce an index entry of the form ⟨*Eastern SNN*⟩ ⟨*Eastern FNN*⟩. No comma is present in the latter form. Furthermore, in the latter two cases, the default output order is Eastern.

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

The way we get these two methods to meet is the reverse output mechanism. One can use the class options described in Section 2.1. One may activate and deactivate reversal with `\ReverseActive` and `\ReverseInactive`. One also may activate reversal for one time with `\RevName`. Using these methods, Western word order can be made to look Eastern and vice-versa. Therefore, the key is to decide the index entry format and choose the name encoding.

The reversing mechanism intelligently swaps name patterns. To illustrate this, we turn off indexing and name formatting and always use full names. A list of Japanese music artists allows us to use `\RevName\Name`... and some creativity:

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

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

Full capitalization of surnames occurs with `\AllCapsActive`, `\AllCapsInactive`, and `\CapName`. These macros are analogous to the reversing macros above and may be used alone or with those macros, e.g. `\CapName\RevName\Name`:

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

The reversing and capitalization macros also work with `\AKA`. They affect only the text, not the index. Whoever wants all-cap forms in the index will have to cap everything manually or modify the macros.

2.5 Other Naming Topics

2.5.1 Listing by Surname

\ReverseCommaActive
\ReverseCommaInactive
\RevComma

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

Jeremy Bentham	Bentham, Jeremy	OK
John Stuart Mill	Mill, John Stuart	OK
John Eriugena	Eriugena John	FAIL
Albertus Magnus	Magnus Albertus	FAIL
Anaximander	Anaximander	OK

There is no way around this "fail" here because that is exactly what we want to happen in the case of Eastern names. It is not possible for this package to be all things to all names, but it tries to be as broad as possible.

2.5.2 Naming Standards

\CapThis

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

Names foreign to English often associate these particles with the ⟨*FNN*⟩ field of `\Name`. Yet these particles are not really first names. Using `\FName` with alternate forenames avoids that issue. See also Section 2.9.3.

2.5.3 Hyphenation

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

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

2.5.4 Accented Names

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

À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	FIRST USE
À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	second use
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ SS	FIRST USE
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ ß	second use
à á â ã ä å æ	ç è é ê ë	ì í î ï ð ñ	FIRST USE
à á â ã ä å æ	ç è é ê ë	ì í î ï ð ñ	second use
ò ó ô õ ö ø	ù ú û ü ý	þ ÿ	FIRST USE
ò ó ô õ ö ø	ù ú û ü ý	þ ÿ	second use
Ă 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 ij Ľ ŀ Ł ł	Ń ń Ň ň Œ œ	Ŕ ŕ Ř ř	FIRST USE
IJ ij Ľ ŀ Ł ł	Ń ń Ň ň Œ œ	Ŕ ŕ Ř ř	second use
Ś Ş ş Š š Ţ ţ Ť ť	Ů ů Ű ű	Ź ź Ż ż Ž ž	FIRST USE
Ś Ş ş Š š Ţ ţ Ť ť	Ů ů Ű ű	Ź ź Ż ż Ž ž	second use

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

```
\usepackage{newunicodechar}
\DeclareTextSymbolDefault{\textlongs}{TS1}
\DeclareTextSymbol{\textlongs}{TS1}{115}
\newunicodechar{ſ}{\textlongs}
\newunicodechar{ā}{\=a}
\newunicodechar{ṃ}{\d m}
```

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

In some cases, indexing accented names only works with xetex and luatex, both of which use fontspec. Using makeindex may require The `-g` option and user-created settings in an `.ist` file may be needed. That goes beyond the scope of this document. One could use `\IndexActive` and `\IndexInactive` to suppress indexing and create manual entries but that would be tedious in non-English documents. Control sequences like `\=a` fail when using makeindex and `gind.ist` because the equal sign is interpreted as a "literal" character, as mentioned by Dan Luecking. I used `\IndexInactive\Name{Ghazāli}\IndexActive` to prevent the index entry "ali" sorted under "Ghaz". Even the manual entry fails in that case.

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

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

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

```
\ifxetex ⟨xelatex text⟩%
\else%
  \ifluatex%
    \ifpdf ⟨lualatex in pdf mode text⟩%
    \else ⟨lualatex in dvi mode text⟩%
    \fi%
  \else%
    \ifpdf ⟨pdflatex text⟩%
    \else ⟨latex text⟩%
    \fi%
  \fi%
\fi
```

2.5.5 Custom Formatting

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

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

This approach synchronizes the "first use" feature in the text and the footnotes, but only suppresses the formatting. It takes advantage of the deep nesting of

`\@makefntext` and a localized `\def` to make a temporary change. A second example puts the mention of first names in the margin if possible:

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

The result produces something like:

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

The forms above do not work with `\PName` and `\AKA`. Consistently use either the suffix mechanism (Section 2.4.1) or see Section 2.6.1 regarding manual entries. If you use the suffix mechanism, you would use the following forms:

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

Vlad III Dracula
Vlad II Dracul

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

The redefinition of `\NamesFormat` given above follows scoping rules. Since I used it in a quote environment, it reverts to normal and now we have: Vlad III Dracula. Later references produce Vlad III. The "Dracula" example was manipulated extensively with the tweaking macros `\ForgetName` and `\SubvertName`. Mixing the comma-delimited suffix and third-parameter affix forms might cause errors.

2.5.6 Disable Formatting

`\NamesActive`
`\NamesInactive`

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

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

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

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

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

Notice that we have two independent cases of "first use" above. That is intended for different sections of the document, like front matter and main matter. It clashes when on the same page or on one nearby.

2.5.7 Tweaks: `\ForgetName` and `\SubvertName`

Perhaps the easiest way to avoid the clashes above are the two macros presented here. They are meant for tweaking text at or near final draft stage. They affect both front matter and main matter.

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

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

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

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

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

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

2.6 Name Variant Macros

2.6.1 Introduction

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

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

Jean sans Peur (Jean the Fearless) reigned as Duke of Burgundy from 1404 to 1419.

The suffix workaround would use `\Name{Jean, sans Peur}` and the author need merely type `\AKA{Jean, sans Peur}{Jean the Fearless}`. See also Section 2.4.1. Using makeindex also requires some manual entries:

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

Perhaps the greatest medieval theologian was Thomas Aquinas (Thomas of Aquino), also known as *Doctor Angelicus*. His name "Aquinas" is not a surname.

2.6.2 `\AKA`

`\AKA`
`\AKA*`

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

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

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

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

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

† This succeeds, but replaces "Leslie Townes" with "Lester T." in the text, while keeping "Hope, Leslie Townes" as the *see* reference in the index.
‡ This form uses the `nocomma` feature.
§ This produces different output, depending on whether `\AKA` or `\AKA*` is used.

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

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

Today we consider Mary Anne Evans and her literary contributions as George Eliot.

If the starred form `\AKA*` is used with the template ⟨*none*⟩⟨*alternate SNN*⟩⟨*alt. names*⟩, it only prints the ⟨*alt. names*⟩. With the same template, `\AKA` prints ⟨*alternate SNN*⟩ followed by ⟨*alt. names*⟩. Section 2.7.2 further illustrates the usefulness of `\AKA`, `\AKA*`, and index tagging.

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

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

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

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

2.6.3 `\PName`

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

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

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

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

2.7 Indexing Macros

2.7.1 `\IndexName`

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

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

`\IndexName` does not index ⟨*alternate names*⟩ unless ⟨*FNN*⟩ are absent, whereupon it treats ⟨*alternate names*⟩ as an affix. If indexing is switched off (see Section 2.7.5), this macro does nothing. It will not create index entries for names that have been used as cross-references.

2.7.2 `\TagName`

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

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

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

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

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

2.7.3 \UntagName

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

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

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

\ForgetName[John]{Smith} \TagName[John]{Smith}% {, the other one} \Name[John]{Smith}...	This is the first reference to John Smith, "the other one" in the index. Notice that it is tweaked.
\ForgetName[John]{Smith} \TagName[John]{Smith}% {, the third} \Name[John]{Smith}...	This refers to John Smith "the third" in the index. It is tweaked again as a "first occurrence."
\SubvertName[John]{Smith} \UntagName[John]{Smith} \Name*[John]{Smith}	This refers to the original John Smith. It is tweaked in order to force a subsequent reference.

Tagging and untagging in this manner requires the author to juggle more info. With more freedom comes more responsibility. Also, if you use \UntagName within a scope, you may need to repeat it after leaving that scope.

2.7.4 \ExcludeName

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

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

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

2.7.5 Indexing Certain Sections

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

2.8 Variant Spellings

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

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

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

2.9 Naming Pattern Reference

2.9.1 Basic Naming

When referring to a name for the first time, the following have the same result. We put the starred form first because its output is always longest:

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

Subsequent references to names differ, based on the macro used:

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

Nicknames and alternate forenames use both the first and third arguments of `\Name`. The index forms are constant even when the print forms change:

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

Different forenames, same surname: Jane Qetsiyah Public	`\Name*[J.Q.]{Public}[Jane Qetsiyah]`
Subsequent name: J.Q. Public	`\Name*[J.Q.]{Public}`
Alternate forename: Janie	`\FName[J.Q.]{Public}[Janie]`

These next examples are the "limited" variants that work with the `comma` option. `\AKA` and `\PName` cannot cross-reference to these forms. Sections 2.4.1 and 2.6.1 address this issue with different solutions.

First Eastern reference: Mao Tse-tung	`\Name*{Mao}[Tse-tung]` `\Name{Mao}[Tse-tung]`
Subsequent refs: Mao Tse-tung	`\Name*{Mao}[Tse-tung]`
Subsequent refs: Mao	`\Name{Mao}[Tse-tung]` `\FName{Mao}[Tse-tung]`

First royal: Louis the Pious	`\Name*{Louis}[the Pious]`
	`\Name{Louis}[the Pious]`
	`\FName{Louis}[the Pious]`

Subsequent refs: Louis the Pious	`\Name*{Louis}[the Pious]`
Subsequent refs: Louis	`\Name{Louis}[the Pious]` `\FName{Louis}[the Pious]`

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

Subsequent refs: Ptolemy I Soter	`\Name*{Ptolemy I}[Soter]`
Subsequent refs: Ptolemy I	`\Name{Ptolemy I}[Soter]` `\FName{Ptolemy I}[Soter]`

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

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

2.9.2 Affixes

Always use a comma to delimit name/affix pairs. `\AKA` and `\PName` will cross-reference these forms. See also Section 2.4.1.

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

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

The next cases depend on the default `nocomma` option, whereby one can use comma suppression to implement forms of ancient, royal and Eastern names. `\AKA` and `\PName` will cross-reference these forms. Cf. the reference to Ptolemy I (Section 2.9.1). Using `\Name{Demetrius, I Soter}` keeps the number with the suffix. To keep the number with the name, use `\Name{Demetrius I, Soter}`.

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

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

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

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

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

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

2.9.3 Particles

The following illustrate the American style of particulate names.

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

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

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

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

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

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

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

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

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.