

The nameauth package

Charles P. Schaum
charles dot schaum at att dot net

v1.26 from 2012/04/24

Abstract

Using the `nameauth` package, an author can encode names according to a name authority. Index entries are consistent with input parameters. In-text formatting of names is predictable and automated, making it easier to go from drafts to a final manuscript. This package mainly supports Western names, with basic features for ancient, royal, and Eastern names.

Contents

1	Introduction	2	2.4.3	Suffix Removal . . .	9
1.1	Typesetting, Indexing, and Design	2	2.4.4	Accented Names . .	10
1.2	Thanks	2	2.4.5	Name Formatting .	12
1.3	Disclaimer	2	2.4.6	Formatting Certain Sections	13
2	Usage	3	2.4.7	Tweaks: <code>\ForgetName</code> and <code>\SubvertName</code>	13
2.1	Package Options	3	2.5	Name Variant Macros . .	14
2.2	Quick Start Guide	4	2.5.1	Introduction and Manual Index Entries	14
2.2.1	Rules of Thumb	4	2.5.2	<code>\AKA</code>	15
2.2.2	Basic Naming	4	2.5.3	<code>\PName</code>	16
2.2.3	Intermediate Naming	5	2.6	Indexing Macros	16
2.2.4	Common Pitfalls	6	2.6.1	<code>\IndexName</code>	16
2.3	Naming Macros	7	2.6.2	<code>\TagName</code>	17
2.3.1	Surnames: <code>\Name</code> and <code>\Name*</code>	7	2.6.3	<code>\UntagName</code>	17
2.3.2	Forenames: <code>\FName</code>	8	2.6.4	<code>\ExcludeName</code> . .	18
2.4	Advanced Naming Topics	9	2.6.5	Indexing Certain Sections	18
2.4.1	Naming Conven- tions	9	2.6.6	Variant Spellings .	18
2.4.2	Hyphenation	9	2.7	Error Handling	19

1 Introduction

Suppose you were working on a collection of essays. Different publishers' permissions may force you to accept variance in the spelling of people's names. You would track and index those names using a name authority. Your index might use abbreviated name forms. This package allows the author to encode names so that the time and cost of an editor and proofreader can be minimized. This could make an author more desirable for publication. Features include:

- Simultaneous printing and indexing of names with fuller, typographically distinct first references.
- Basic handling of variant names.
- Printing alternate forenames, yet indexing a canonical form.
- Basic formatting and indexing tasks.
- Automatic addition of tags to names in indexes.

1.1 Typesetting, Indexing, and Design

This package depends on `etoolbox` and `xparse`. It has been tested with `latex`, `lualatex`, `pdflatex` and `xelatex`, as well as `makeindex` and `texindy`. This file was typeset with `pdflatex` and `makeindex`. The default options and design of this package try to minimize keystrokes or trade extra work for benefits in formatting and consistency. The `.dtx` file is also a good source of information.

1.2 Thanks

Thanks to MARC VAN DONGEN, ENRICO GREGORIO, PHILIPP STEPHANI, HEIKO OBERDIEK, UWE LUECK, and ROBERT SCHLICHT 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 and the sagacious advice of Robert enabled the routines to function with the `microtype` package.

1.3 Disclaimer

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

2 Usage

2.1 Package Options

The defaults are `mainmatter`, `smallcaps`, `nocomma`, and `index`. With these options, the package begins immediately to format the first occurrences of names with small caps using a longer version of the name if one exists. It suppresses commas between surnames and suffixes and it indexes all names in place. The following class options alter this behavior.

<code>mainmatter</code>	Use special typesetting for the first occurrence of a name, starting at the beginning of a document.
<code>frontmatter</code>	Suppress the special typesetting <i>before</i> the invocation of <code>\NamesActive</code> . This option fits well, e.g., with front matter from a contributor who may not intend the same formatting and emphasis found in the main matter. The “first use” mechanism used with this option works independently of the “first use” mechanism after the invocation of <code>\NamesActive</code> . Indexing and aliasing features remain consistent. See also Section 2.4.6.
<code>smallcaps</code>	Set the first use of a name in small caps.
<code>italic</code>	Italicize the first occurrence of a name.
<code>boldface</code>	Set the first use of a name in boldface.
<code>noformat</code>	This option suppresses name formatting <i>after</i> the invocation of <code>\NamesActive</code> . It does not prevent formatting like the <code>frontmatter</code> option. Additional formatting options are discussed in Section 2.4.5, which works differently than the methods in Section 2.4.6.
<code>nocomma</code>	Suppress commas between surnames and suffixes, following modern styles like <i>Chicago Manual of Style</i> . See also Section 2.4.3.
<code>comma</code>	Retain commas between surnames and suffixes. This imposes limits on the use of certain macros.
<code>index</code>	Create index entries in place with the names.
<code>noindex</code>	Prevent indexing before the invocation of the macro <code>\IndexActive</code> . See also Section 2.6.5.

2.2 Quick Start Guide

This gets one using about 60% of the package features right away.

2.2.1 Rules of Thumb

- Use `\frenchspacing`.
- The indexed forms of the names remain the same if you follow the macro input charts consistently.
- Examine index entries to find and correct mistakes.
- Trade work for consistency.
- Use just the macros you *need*.

2.2.2 Basic Naming

The macro used most often is `\Name`. When referring to a name for the first time, the following macros all have the same result:

1. First reference: JOHN SMITH
`\Name*[John]{Smith}` or
`\Name[John]{Smith}` or
`\FName[John]{Smith}`
2. First mononym reference: PLATO
`\Name*{Plato}` or
`\Name{Plato}` or
`\FName{Plato}`

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

1. Subsequent full name: John Smith
`\Name*[John]{Smith}`
2. Subsequent surname: Smith
`\Name[John]{Smith}`
3. Subsequent forename: John
`\FName[John]{Smith}`
4. Subsequent mononym: Plato
`\Name*{Plato}` or
`\Name{Plato}` or
`\FName{Plato}`

These next examples work with or without the `comma` option. `\AKA` and `\PName` cannot cross-reference these forms. Page 5, Section 2.4.3, and Section 2.5.1 address this issue with different solutions.

1. First Eastern ref: MAO TSE-TUNG
`\Name*{Mao}[Tse-tung]` or
`\Name{Mao}[Tse-tung]`

2. Subsequent references: Mao Tse-tung
`\Name*{Mao}[Tse-tung]`
3. Subsequent references: Mao
`\Name{Mao}[Tse-tung]` or
`\FName{Mao}[Tse-tung]`
4. First royal: LOUIS THE PIOUS
`\Name*{Louis}[the Pious]` or
`\Name{Louis}[the Pious]` or
`\FName{Louis}[the Pious]`
5. Subsequent refs: Louis the Pious
`\Name*{Louis}[the Pious]`
6. Subsequent references: Louis
`\Name{Louis}[the Pious]` or
`\FName{Louis}[the Pious]`
7. First ancient: PTOLEMY I SOTER
`\Name*{Ptolemy I}[Soter]` or
`\Name{Ptolemy I}[Soter]` or
`\FName{Ptolemy I}[Soter]`
8. Subsequent refs: Ptolemy I Soter
`\Name*{Ptolemy I}[Soter]`
9. Subsequent refs: Ptolemy I
`\Name{Ptolemy I}[Soter]` or
`\FName{Ptolemy I}[Soter]`
10. First royal: HENRY VIII
`\Name*{Henry}[VIII]` or
`\Name{Henry}[VIII]` or
`\FName{Henry}[VIII]`
11. Subsequent refs: Henry VIII
`\Name*{Henry}[VIII]`
12. Subsequent references: Henry
`\Name{Henry}[VIII]` or
`\FName{Henry}[VIII]`

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

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

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

2.2.3 Intermediate Naming

Always use a comma to delimit suffixes, followed by a space or thin space. `\AKA` and `\PName` will cross-reference these forms. See also Section 2.4.3. Macros that are too wide for the column are split across lines using the comment token, but need not be so in the text.

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

These cases depend on the default `nocomma` option, whereby one can use comma suppression to implement forms of ancient, royal and Eastern names. By avoiding the final optional parameter of `\Name` and friends, `\AKA` and `\PName` will cross-reference these forms.

`\Name{Ptolemy I}[Soter]` (see above) keeps the number with the name. Below, the form `\Name{Demetrius, I Soter}` keeps the number with the sobriquet. To keep the number with the name, use `\Name{Demetrius I, Soter}`.

1. First reference: FRANCIS I
`\Name*[Francis, I]` or
`\Name{Francis, I}` or
`\FName{Francis, I}`

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

Particulate names contain prepositions that refer to an ancestral place or name. The following illustrate the American style of particulate names.

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

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

The following examples illustrate the Continental style of particulate names. Forms 1–4 keep the particles with the names in the index. Forms 5–7 omit the particles in the index by using alternate forenames. Long commands are split using the comment token, but need not be so in the text.

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

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

2.2.4 Common Pitfalls

Avoiding the following pitfalls will save much time and frustration:

1. Mixing sobriquets with modern forms fails due to the “alternate name” feature that replaces the first with the final optional parameters, e.g.:
`\Name[First]{Ancient}[Sobriquet]
\Name[King]{Number}[Sobriquet]`
2. Using `\AKA` and `\PName` with forms that use the final optional parameter of `\Name` and friends will fail unless one uses comma-delimited suffixes. See Section 2.4.3 and Section 2.5.2.
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. `\Name` can shorten a reference even if you use the full name in the text. Do not depend on the name you see when you are typing. Depend on the form of the macro, whether `\Name`, `\Name*`, or `\FName`.
5. Unbalanced or incorrect `{` braces `}` and `[` brackets `]`, putting a star `*` after `\FName`, and forgetting periods in initials create errors that can be difficult to track. Leading spaces in macro arguments will create incorrectly sorted index entries. The results are usually a failure to finish scanning the document or a spurious index entry.

2.3 Naming Macros

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

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

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

From now on we will abbreviate `⟨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:

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

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.

As long as the “main” `⟨FNN⟩` are constant in Western-style names, the “alternate names” field can vary, yet the index entries will be constant. Section 2.2 contains examples that illustrate this ability to swap out forenames. This is useful when dropping out references to middle initials in `\FName`. Alternate forenames only get printed in subsequent occurrences of `\Name*` and `\FName`. The surname argument is *always* printed in `\Name*` and `\Name`.

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 2.4.3. The following method is the only one that works with the `comma` option. Valid “sobriquet” forms are:

$\langle FNN \rangle$	$\langle SNN \rangle$	$\langle Alternate Names \rangle$	Result	Function
(none)	Henry	VIII	HENRY VIII	<code>\Name{Henry}[VIII]</code>
(none)	Henry	VIII	Henry	<code>\Name{Henry}[VIII]</code>
(none)	Chiang	Kai-shek	CHIANG KAI-SHEK	<code>\Name{Chiang}[Kai-shek]</code>
(none)	Chiang	Kai-shek	Chiang	<code>\Name{Chiang}[Kai-shek]</code>

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

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

2.3.2 Forenames: `\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. `\FName` is intended for Western-style names. The syntax is basically the same:

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

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

$\langle FNN \rangle$	$\langle SNN \rangle$	$\langle Alternate Names \rangle$	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. For nicknames use the “alternate names” option. 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 2.4.3). These include names like J.D. ROCK III and CHARLES V:

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

`\FName` suppresses extra periods if a forename with initials occurs at the end of a sentence, as in the plot-line “who shot J.D.” See also Section 2.4.3.

2.4 Advanced Naming Topics

2.4.1 Naming Conventions

`\CapThis` English names with the particles *de*, *de la*, *d'*, *von*, *van*, and *ten* generally keep them with the last name, using varied capitalization. *Le*, *La*, and *L'* are capitalized unless preceded by *de*. In English, these particles go in the $\langle SNN \rangle$ 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) might disagree. For the present, at the start of a sentence, use `\textsc{Du Cange}\IndexName{du Cange}\SubvertName{du Cange}`.

Names foreign to English usually put these particles in the $\langle FNN \rangle$ field of `\Name`. Yet these particles are not first names. Using `\FName` with alternate forenames avoids the particles. See the examples on page 6.

2.4.2 Hyphenation

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

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

2.4.3 Suffix Removal

`\Name`—not `\Name*`—truncates comma-delimited suffixes from last names. For example, it prints the name OSKAR HAMMERSTEIN II the first time and Hammerstein thereafter. One must always use a comma to activate this, e.g., `\Name[Oskar]{Hammerstein, II}`. You should use a space or a thin space after the comma in order to separate the suffix from the surname.

Again, **the comma is not optional with suffixes**. It is how this feature works. More than one comma in the $\langle SNN \rangle$ 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 combinations:

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

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

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

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

Even though suffix and sobriquet features look like they produce the same *output* in the body text using the `nocomma` option, they are internally *different*. They will not respect each other regarding ‘‘first use,’’ although they will (tentatively) cooperate in the index. Use each approach consistently. An example of ‘‘dangerous’’ use of these features occurs in Section 2.4.5. The `comma` option will cause these forms above to have commas and behave differently.

2.4.4 Accented Names

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
Ǻ ǻ Ǽ Ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	FIRST USE
Ǻ ǻ Ǽ Ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	Ǿ ǿ ǿ ǿ ǿ ǿ ǿ	second use
Ĳ ĳ Ĵ ĵ Ķ Ŀ	Ń ņ Ň ň Œ œ	Ř ř Ŕ ŕ	FIRST USE
Ĳ ĳ Ĵ ĵ Ķ Ŀ	Ń ņ Ň ň Œ œ	Ř ř Ŕ ŕ	second use
Š š Š š Ť ť Ţ ţ	Ů ů Ű ű	Ž ž Ž ž Ž ž	FIRST USE
Š š Š š Ť ť Ţ ţ	Ů ů Ű ű	Ž ž Ž ž Ž ž	second use

Please remember that if you use accented names, you may prefer using `xindy` (or `texindy`) on the `.idx` file. It simplifies the whole process and I recommend it.

With `makeindex` there are a number of solutions, but some depend on the availability of the `-g` option and user-created settings in an `.ist` file whose description

goes beyond the scope of this document. You also could use `\IndexActive` and `\IndexInactive` to suppress indexing, then create manual entries. For non-English documents, however, that solution is not optimal.

One may add control sequences to names, thanks to the help of Robert Schlicht. The next example adds more accented characters under `inputenc`/`fontenc`:

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

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

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

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

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

I used the following in the text itself to allow for conditional processing that helps one 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 <pdfplatex text>%
    \else <latex text>%
    \fi%
  \fi%
\fi
```

2.4.5 Name Formatting

`\NamesFormat` The first instance of a name is formatted with `\NamesFormat`. In addition to using the class options, one can redefine `\NamesFormat`. 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. The code below gives us:

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

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

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

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

Vlad III Dracula
Vlad II Dracul

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

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

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

The “Dracula” example above was manipulated extensively. Mixing the sobri-
quet and suffix forms otherwise may cause errors to “bite.” Puns aside, Vlad III
was a complex historical figure, more so than the character of BRAM STOKER.

2.4.6 Formatting Certain Sections

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

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

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

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

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

Notice that we have two cases of “first use.” That works fine in different parts
of the document, but not within the same part. Avoid using `\NamesInactive` to
turn off formatting in footnotes because that can produce undesirable results like
the example above. Fixing that would require use of the tweaking macros in both
body text and footnotes. Instead, use the example for suppressing formatting in
Section 2.4.5 and apply tweaks like `\ForgetName` in the body text as needed.

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

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

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

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

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

2.5 Name Variant Macros

2.5.1 Introduction and Manual Index Entries

The macro `\AKA` deals with pseudonyms, stage names, *noms de plume*, etc. With this macro we move into a hybrid naming/indexing area of functionality. `\Name` and friends produce index entries compatible with manual index entries since version 0.9. If needed, the `.idx` file can be a helpful reference when linking manual entries with `nameauth` entries.

Before we examine `\AKA` in detail, we should cover the manual solution that `\AKA` helps to automate. This remains the only solution for name forms that use the final optional parameter of `\Name` and friends, especially if one chooses the `comma` option. It also is the only option for using special formatting in the text.

```
\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}` but be careful; see Section 2.4.5. 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.
```

This case must be handled manually due to the italic text formatting.

2.5.2 \AKA

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

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

The ⟨FNN⟩ and ⟨SNN⟩ arguments *do not include* the final optional argument of \Name to avoid ambiguity. \AKA will not create a *see* reference to a name with a sobriquet or to an Eastern name unless one uses the suffix feature mentioned previously. This macro is designed with Western names in mind.

\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.

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.

\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*⟩.

If the starred form \AKA* is used with the template ⟨*none*⟩{⟨*alternate SNN*⟩}[⟨*alt. names*⟩], it only prints the ⟨*alt. names*⟩. With the same template, \AKA prints ⟨*alternate SNN*⟩ followed by ⟨*alt. names*⟩. Section 2.6.2 offers a complete presentation of related features. 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 cross-references generated by \AKA and \AKA* cannot be used with other macros in this package. See also Section 2.7.

In certain cases, the alternate name might need to be indexed with page numbers. Do not use \AKA if that is so. Use \Name for both the main and the alternate names. Then create manual cross-references with \index, e.g.:

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

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

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

2.5.3 `\PName`

`\PName` `\PName` is a “convenience macro” that sacrifices flexibility for simplicity. It does not implement `\AKA*` and it works best with Western-style names. It calls `\Name` or `\Name*` and prints a Western-style “main” name. It then always calls `\AKA` with a full alternate name and prints it in parentheses. The syntax is:

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

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} compared to
\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} and
\PName{Voltaire}[François-Marie]{Arouet}
Print VOLTAIRE (François-Marie Arouet) the first time it appears.
Later, print Voltaire (François-Marie Arouet).
```

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 Indexing Macros

2.6.1 `\IndexName`

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

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


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

2.6.2 `\TagName`

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

`\TagName[\langle FNN \rangle]{\langle SNN \rangle}[\langle alternate names \rangle]{\langle tag \rangle}`

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

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

`\TagName` causes tags in the index entries to be inserted automatically from the point of invocation. `\AKA` prints the name and the sobriquet, while `\AKA*` only prints the sobriquet, One could do similarly with emperors, kings, etc.

2.6.3 `\UntagName`

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

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

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

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

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

2.6.4 `\ExcludeName`

`\ExcludeName` This prevents `\Name`, etc. from both formatting and indexing a specific name, but *only if that name has not been used*. See also Section 2.7. The syntax is:

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

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

2.6.5 Indexing Certain Sections

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

2.6.6 Variant Spellings

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

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

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

2.7 Error Handling

Except for the indexing macros, most of the macros in this package print any erroneous arguments in the text without formatting. Unless there is a syntax error, all macros emit meaningful warnings. The convenience macros do not emit warnings; however, their component macros do. For example, `\PName` produces warnings via `\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. Some warnings, especially in the case of indexing macros, indicate that the macro produced no output.

Examine the warning messages and check the index in order to find possible errors. 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.

Change History

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

Index

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

A		Gregorio, Enrico 2	<code>\NamesActive</code> 13
<code>\AKA</code> 15		Gregory I, pope 17	<code>\NamesFormat</code> 12
<code>\AKA*</code> 15		Gregory the Great	<code>\NamesInactive</code> 13
Arouet, François-Marie		see Gregory I	
see Voltaire			
C		H	O
<code>\CapThis</code> 9		Hammerstein, Oskar, II 9	Oberdiek, Heiko 2
Carnap, Rudolph 13		Harnack, Adolf 6	P
Charles the Bald, emperor 7, 8		Henry VIII, English king 4, 8	Patton, George S., Jr. 5
Charles V, emperor 8		I	Planck, Max 8
Chiang Kai-shek 8		<code>\IndexActive</code> 18	Plato 4
Cicero, M.T. 7, 8		<code>\IndexInactive</code> 18	<code>\PName</code> 16
Clemens, Samuel L.		<code>\IndexName</code> 16	Ptolemy I Soter, king 4
see Twain, Mark			Public, J.Q. 5
Confucius 7, 8		J	R
D		Jean sans Peur, duke 14	Rambam see also
de la Mare, Walter 5, 6, 9		Jean the Fearless	Maimonides, 16
Demetrius I Soter, king 5		see Jean sans Peur	Rock, J.D., III 8
<i>Doctor Angelicus</i> see			S
Thomas Aquinas		K	Schlicht, Robert 2, 11
Dongen, Marc van 2		King, Martin Luther, Jr. 9	Smith, John 4, 18
Du Bois, W.E.B. 18		L	Smith, John, the other one 18
du Cange 9		Leo I, pope 17	Smith, John, the third 18
du Fresne, Charles		Leo the Great see Leo I	Snel van Royen, Rudolph 16
see du Cange		Louis the Pious, emperor 4	Snel van Royen, Willebrord 16
E		Louis XIV, French king 10	Snellius see Snel van
Einstein, Albert 7, 8		Lueck, Uwe 2	Royen, Rudolph, see Snel van
Eliot, George 15		Luecking, Dan 11	Royen, Willebrord
Evans, Mary Anne		M	Stephani, Philipp 2
see Eliot, George		Maimonides 16	Stoker, Bram 13
<code>\ExcludeName</code> 18		Malebranche, Nicolas 13	Strietelmeier, John 9
F		Mao Tse-tung 4	<code>\SubvertName</code> 14
<code>\FName</code> 8		Moses ben-Maimon	Sullenberger, Chesley B., III 8
<code>\ForgetName</code> 13		see Maimonides	Sun King see Louis XIV
Francis I, French king 5		N	Sun Yat-sen 5, 10
G		<code>\Name</code> 7	T
Goethe, Johann Wolfgang von 6		<code>\Name*</code> 7	<code>\TagName</code> 17

Thomas Aquinas	14	Twain, Mark	16	Vlad III Dracula
				Vlad II Dracul . . .
				12, 13
				Vlad III Dracula . .
				12, 13
				Vlad the Impaler
				<i>see</i>
Thomas of Aquino	<i>see</i>			Vlad III Dracula
Thomas Aquinas		Vlad Țepeș	<i>see</i>	Voltaire
				16