

The nameauth package

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

v0.8 from 2012/01/29

Abstract

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

1 Introduction

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

1.1 Design Decisions

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

1.2 Thanks

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

2 Usage

2.1 Package Options

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

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

2.2 Quick Start Guide

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

1. The indexed forms of the names always remain the same.
2. Always use the same form of reference for, e.g., `\Name[John]{Smith}` or `\Name{Louis}[XIV]`, otherwise point 1 will become false.
3. Typing longer name references allows you to rearrange text and let L^AT_EX figure out first references and name forms. This package trades some work for consistency.
4. Checking against the index will help you debug your `\Name` references, *inter alia*.
5. Start using the macros you *need*, then work from there.

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

† Using this form can cause unwanted results if you rearrange text.

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

‡ The alternate names are ignored in this case.

¶ “Unofficial” variants are neither shown nor encouraged.

I want to...	I need to use (for example)...
Print a full Western name followed by a lesser-known name in parentheses (first reference)	<code>\PName*[Bob]{Hope}[Leslie Townes]{Hope}</code> or <code>\PName[Bob]{Hope}[Leslie Townes]{Hope}</code>
Print a full Western name followed by a lesser-known name in parentheses (subsequent reference)	<code>\PName*[Bob]{Hope}[Leslie Townes]{Hope}</code>
Print a surname only, followed by a full lesser-known name in parentheses (subsequent reference)	<code>\PName[Bob]{Hope}[Leslie Townes]{Hope}</code>
Print a mononym followed by a lesser-known name in parentheses	<code>\PName*{Prince}[Prince Rogers]{Nelson}</code> or <code>\PName{Prince}[Prince Rogers]{Nelson}</code>
Print a name, then another name that only cross-references it in the index	a tribute to <code>\Name{Ari Up}</code> , born <code>\AKA{Ari Up}[Arianne]{Forster}...</code>

These (and possibly other combinations) will likely fail:

1. `\Name[First]{Ancient}[Sobriquet]` and `\Name[King]{Number}[Sobriquet]`
2. `\Name{Ancient}[Sobriquet]` `\AKA{Ancient Sobriquet}[OtherFirst]{OtherLast}`
3. `\Name{Queen}[Number]` `\AKA{Queen Number}[OtherFirst]{OtherLast}`
4. `\Name{EastFamily}[EastFirst]` `\AKA{EastFamily EastFirst}[OtherFirst]{OtherLast}`
5. E.g. `\Name[] {Kreskin}[The Amazing]` (`\AKA{Kreskin}[Joseph]{Kresge}`)
6. Using `\PName` in the same way as `\AKA` above.
7. Using `\Name` with a lot of people having the same surname or mononym and different numbers without disambiguation in the text.
8. Getting the pairing of { and } or [and] messed up.

2.3 Basic Macros

2.3.1 `\Name` and `\Name*`

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

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

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

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

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

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

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

Here are some more examples of use:

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

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

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

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

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

`\Name{Antiochus IV}{Epiphanes}`

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

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

2.3.2 Suffix Removal

`\Name` and `\Name*` have an additional difference. `\Name` will truncate the comma and suffizes from subsequent occurrences of last names. For example, it prints the name OSKAR HAMMERSTEIN, II the first time and Hammerstein thereafter.

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

<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.

2.3.3 Tabular Summary of `\Name`

We begin by recapitulating the original uses: Usual Western forenames and surnames, persons with a mononym, an abbreviated name in the index and a fuller name in the text, and a mononym plus sobriquet option:

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

Different results can be achieved when the options are applied creatively. In some cases, the sobriquet option can serve for lists of royalty and references to them. In others, this use will fail:

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

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

<i>FNN</i>	<i>SNN</i>	Alternate Names	Result	Function
(none)	Chiang	Kai-shek	CHIANG KAI-SHEK	\Name{Chiang}[Kai-shek]
(none)	Chiang	Kai-shek	Chiang	\Name{Chiang}[Kai-shek]
(none)	Sun	Yat-sen	SUN YAT-SEN	\Name{Sun}[Yat-sen]
(none)	Sun	Yat-sen	Sun	\Name{Sun}[Yat-sen]

Here is a general table of options. Sobriquets and suffixes do mix, but not well. Therefore their use together is not shown.

<i>FNN</i>	<i>SNN</i>	Alternate Names	Result
Forename	SurnameA, Suffix	Othername	OTHERNAME SURNAMEA, SUFFIX (\Name)
Forename	SurnameA, Suffix	Othername	Othername SurnameA, Suffix (\Name*)
Forename	SurnameA, Suffix	Othername	SurnameA (\Name)
Forename	SurnameB, Suffix	(none)	FORENAME SURNAMEB, SUFFIX (\Name)
Forename	SurnameB, Suffix	(none)	Forename SurnameB, Suffix (\Name*)
Forename	SurnameB, Suffix	(none)	SurnameB (\Name)

2.4 Advanced Features

2.4.1 Alternate forenames in the Text

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

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

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

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

The alternate forenames *override* the *FNN* in the text. This is *not* true with sobriquets, which are *appended* to *SNN*. The difference in the two is whether the *FNN* argument exists. Alternate forenames do not get printed in subsequent

occurrences of `\Name`. They *do* get printed in subsequent occurrences of `\Name*`. The surname argument is *always* printed. There is, however, the possibility that you might refer to a person with a nickname only. The next macro facilitates this.

2.4.2 `\IndexName`

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

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

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

```
\Name[C.F.W.]{Walther}[Carl Ferdinand Wilhelm] was a major
figure in nineteenth-century American Lutheran history.
\Name[C.F.W.]{Walther} used his full name when signing
official documents. To his family, friends, and close
colleagues, however, he was simply
Ferdinand\IndexName[C.F.W.]{Walther}.
```

CARL FERDINAND WILHELM WALTHER was a major figure in nineteenth-century American Lutheran history. Walther used his full name when signing official documents. To his family, friends, and close colleagues, however, he was simply Ferdinand.

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

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

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

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

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

`\IndexName` may be helpful in a pinch, but it lacks certain safeguards for handling pseudonyms that the following macros introduce.

2.4.3 \PName and \AKA: Pen Names

This section deals with pen names that function differently than the sobriquet feature and apart from it. I have designed these macros to preclude the alternate name and sobriquet features of \Name for the following reasons:

1. It only makes sense for \PName and \AKA to have the better-known name come first, then the lesser-known name.
2. Having the names in that order produces an ambiguous argument list when implementing sobriquets. Making the list unambiguous by using a mandatory argument often would require the author to type an empty set of braces.
3. The more names and details that get packed into any given macro, the more confusing it is to use. That level of inefficiency defeats the purpose of this package, which is to save typing, yet have standardized names.
4. A number of ancient pseudonyms require special typesetting that must be handled manually.

The workaround for a name using a sobriquet is simply:

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

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

A more complicated example is:

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

This also means, unfortunately, that pen names used with Eastern names will have to be indexed manually.

\PName Pseudonyms, *noms de plume*, epithets, and monikers can be handled with two macros: \PName and \AKA. The macro \PName takes a well-known name, followed by as lesser-known name. Like \Name, \PName has starred and unstarred forms, but those forms only operate on the main name that is indexed, not on the second, lesser-known name. The syntax is:

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

The following examples should make clear its use:

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

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

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

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

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

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

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

\AKA only prints the pseudonym, *not* the indexed name. The macro also will prevent the double-printing of a period after an abbreviation. Here is an example:

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

Today we consider Mary Anne Evans and her literary contributions as
GEORGE ELIOT.
```

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

Pseudonyms cannot be indexed with page numbers. They only work as “see” cross-references. **\Name** will print its arguments and emit a warning if you give a pen name. If you want to index a pseudonym with page numbers, use **\Name** for both the main name and the pseudonym, then manually cross-reference them, e.g.:

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

\AKA will not create multiple instances of a cross-reference. This prevents bogus multiple cross-references in the index, but it also excludes the special case where one moniker applies to multiple people, e.g.: Willebrord Snel van Royen (Snellius) and his son RUDOLPH SNEL VAN ROYEN (Snellius). Do not use \IndexName in this case. One must add a manual index entry as a workaround:

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

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

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

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

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

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

2.5 Accented Names

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

À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	FIRST USE
À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	second use
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ ß	FIRST USE
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ ß	second use
À Á Â Ã Ä Å Æ	Ç È É Ê Ë	Ì Í Î Ï Ð Ñ	FIRST USE
à á â ã ä å æ	ç è é ê ë	ì í î ï ð ñ	second use
Ò Ó Ô Õ Ö Ø	Ù Ú Û Ü Ý	Þ ÿ	FIRST USE
ò ó ô õ ö ø	ù ú û ü ý	þ ÿ	second use
Ă Ǻ Ȧ Ȧ Ć ć Č č	Ď ě Đ đ Ě ě Ě ě	Ǧ ǧ Ĩ ĩ	FIRST USE
Ă Ǻ Ȧ Ȧ Ć ć Č č	Ď đ Đ đ Ě ě Ě ě	Ǧ ǧ Ĩ ĩ	second use
IJ iJ L l Ł ł	Ń ń Ņ ņ Œ œ	Ř ř Ť ť	FIRST USE
IJ ij L l Ł ł	Ń ń Ņ ņ Œ œ	Ř ř Ť ť	second use
Š š Š š Ţ ţ Ţ ţ	Ů ů Ű ű	Ž ž Ž ž Ž ž	FIRST USE
Š š Š š Ţ ţ Ţ ţ	Ů ů Ű ű	Ž ž Ž ž Ž ž	second use

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

```

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

```

Please remember that the appropriate font packages, such as `lmodern` or the `TeX Gyre` fonts, are needed to obtain some TS1 glyphs. Also there may be points where `pdflatex` will accept the input (e.g., Ghāzali), but `makeindex` will have problems with that. In “normal” operation (I have used `article` and `memoir`) these control sequences appear to work. Yet in generating this style file, that was not the case. You may need to use manual index entries in those cases.

2.6 Name Formatting

2.6.1 Font Attributes

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

```

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

```

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

```

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

```

Vlad III Dracula
Vlad II Dracul

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

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

Let me again stress that with names like Vlad II Dracul and Vlad III Dracula, you cannot use either `\PName` or `\AKA`. Attempting to do so will fail. You must use as a guide the examples given above for Jean sans Peur and Thomas Aquinas.

2.7 Formatting Certain Sections

`\NamesActive` Using the `frontmatter` option deactivates formatting until `\NamesActive` occurs.
`\NamesInactive` Another macro, `\NamesInactive`, will deactivate formatting again. These two macros toggle formatting on and off. The mechanism works in a complementary, yet independent manner. It can be used throughout the document.

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

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

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

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

2.8 Tweaks:\ForgetName

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

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

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

3 Implementation

3.1 Class Options and Required Packages

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

These boolean values control whether or not formatting of first entries occurs and whether or not a name ends with a period.

```
3 \DeclareOption{mainmatter}{\@nameauth@DoFormattrue}
4 \DeclareOption{frontmatter}{\@nameauth@DoFormatfalse}
5 \DeclareOption{smallcaps}{\newcommand{\NamesFormat}{\scshape}}
6 \DeclareOption{italic}{\renewcommand{\NamesFormat}{\itshape}}
7 \DeclareOption{boldface}{\renewcommand{\NamesFormat}{\bfseries}}
8 \DeclareOption{noformat}{\renewcommand{\NamesFormat}{}}
9 \ExecuteOptions{smallcaps,mainmatter}
10 \ProcessOptions\relax
```

```

11 \RequirePackage{etoolbox}
12 \RequirePackage{xparse}

```

3.2 Internal Macros

`\@nameauth@CleanName`

```

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

```

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

`\@nameauth@Zapsuffix`

```

14 \newcommand{\@nameauth@Zapsuffix}[1]{%
15     \def\@@Zapsuffix##1,##2*{##1}%
16     {\@@Zapsuffix#1,*}%
17 }

```

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

`\@nameauth@CheckDot`

```

18 \def\@nameauth@CheckDot{\futurelet\@token\@nameauth@EvalDot}

```

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

`\@nameauth@EvalDot`

```

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

```

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

`\@nameauth@TestDot`

```

20 \newcommand*{\@nameauth@TestDot}[1]{%
21     \def\TestDot##1.\TestEnd##2\TestStop{\TestPunct{##2}}%
22     \def\TestPunct##1{\ifx\TestPunct##1\TestPunct\else\@nameauth@Puncttrue\fi}%
23     \@nameauth@Punctfalse%
24     \TestDot#1\TestEnd.\TestEnd\TestStop%
25 }

```

While `\@nameauth@CheckDot` looks *ahead* for a period, `\@nameauth@TestDot`—based on a snippet by Uwe Lueck—checks for a terminal period in the name passed to it, ignoring medial periods. This string test is compatible with the `microtype` package.

`\@nameauth@FmtName`

```

26 \DeclareDocumentCommand\@nameauth@FmtName{s m}%
27 {%
28     \@nameauth@TestDot{#2}%
29     \IfBooleanTF{#1}{#2}{\bgroup\NamesFormat{#2}\egroup}%
30 }

```

`\@nameauth@FmtName` is where the first occurrences of a name are formatted. Notice how `\NamesFormat` sits between a `\bgroup` and an `\egroup` to localize the font change. It is adjacent to the brackets that enclose the second parameter, allowing one to hook into the name formatting procedure.

3.3 User Interface Macros

`\Name`

```

31 \DeclareDocumentCommand\Name{s o m o}%
32 {%
33     \@nameauth@Punctfalse%
34     \IfValueTF{#2}%
35     {\IfValueTF{#4}{\def\@ForeNames{#4}}{\def\@ForeNames{#2}}}%
36     \ifcsname\@nameauth@CleanName{#2#3!PN!}\endcsname%
37         \@nameauth@FmtName*{#2 #3}%
38         \PackageWarning{nameauth}%
39         {You cannot create a page reference from the pen name: #2 #3.}%
40     \else%
41         \if@nameauth@DoFormat%
42             \ifcsname\@nameauth@CleanName{#2#3!MN!}\endcsname%
43             \IfBooleanTF{#1}%
44                 {\@nameauth@FmtName*{\@ForeNames{\space}#3}\index{#3,{\space}#2}}%
45                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}\index{#3,{\space}#2}}}%
46             \else%
47                 \csgdef{\@nameauth@CleanName{#2#3!MN!}}{}%
48                 \@nameauth@FmtName*{\@ForeNames{\space}#3}\index{#3,{\space}#2}%
49             \fi%
50         \else%
51             \ifcsname\@nameauth@CleanName{#2#3!NF!}\endcsname%
52             \IfBooleanTF{#1}%
53                 {\@nameauth@FmtName*{\@ForeNames{\space}#3}\index{#3,{\space}#2}}%
54                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}\index{#3,{\space}#2}}}%
55             \else%
56                 \csgdef{\@nameauth@CleanName{#2#3!NF!}}{}%
57                 \@nameauth@FmtName*{\@ForeNames{\space}#3}\index{#3,{\space}#2}%
58             \fi%
59         \fi%
60     \fi}%
61 {\IfValueTF{#4}
62     {\ifcsname\@nameauth@CleanName{#3#4!PN!}\endcsname%
63         \@nameauth@FmtName*{#3 #4}%
64         \PackageWarning{nameauth}%

```

```

65         {You cannot create a page reference from the pen name: #3 #4.}%
66     \else%
67         \if@nameauth@DoFormat%
68             \ifcsname\@nameauth@CleanName{#3#4!MN!}\endcsname%
69             \IfBooleanTF{#1}%
70                 {\@nameauth@FmtName*{#3 #4}\index{#3 #4}}%
71                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}\index{#3 #4}}%
72             \else%
73                 \csgdef{\@nameauth@CleanName{#3#4!MN!}}{}%
74                 \@nameauth@FmtName{#3 #4}\index{#3 #4}%
75             \fi%
76         \else%
77             \ifcsname\@nameauth@CleanName{#3#4!NF!}\endcsname%
78             \IfBooleanTF{#1}%
79                 {\@nameauth@FmtName*{#3 #4}\index{#3 #4}}%
80                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}\index{#3 #4}}%
81             \else%
82                 \csgdef{\@nameauth@CleanName{#3#4!NF!}}{}%
83                 \@nameauth@FmtName*{#3 #4}\index{#3 #4}%
84             \fi%
85         \fi%
86     \fi}%
87     {\ifcsname\@nameauth@CleanName{#3!PN!}\endcsname%
88         \@nameauth@FmtName*{#3}%
89         \PackageWarning{nameauth}%
90         {You cannot create a page reference from the pen name: #3.}%
91     \else%
92         \if@nameauth@DoFormat%
93             \ifcsname\@nameauth@CleanName{#3!MN!}\endcsname%
94             \IfBooleanTF{#1}%
95                 {\@nameauth@FmtName*{#3}\index{#3}}%
96                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}\index{#3}}%
97             \else%
98                 \csgdef{\@nameauth@CleanName{#3!MN!}}{}%
99                 \@nameauth@FmtName{#3}\index{#3}%
100             \fi%
101         \else%
102             \ifcsname\@nameauth@CleanName{#3!NF!}\endcsname%
103             \IfBooleanTF{#1}%
104                 {\@nameauth@FmtName*{#3}\index{#3}}%
105                 {\@nameauth@FmtName*{\@nameauth@Zapsuffix{#3}}\index{#3}}%
106             \else%
107                 \csgdef{\@nameauth@CleanName{#3!NF!}}{}%
108                 \@nameauth@FmtName*{#3}\index{#3}%
109             \fi%
110         \fi%
111     \fi}%
112 }%
113 \if@nameauth@Punct\expandafter\@nameauth@CheckDot\fi%
114 }

```


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

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

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

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

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

Where longer and shorter versions of names are printed, there the star parameter controls those outcomes.

The use of `\expandafter` before `\@nameauth@CheckDot` works with the other use of `\expandafter` mentioned with `\@nameauth@EvalDot` above to move past the closing brace and fetch the period as lookahead. That is only done when the check for a terminal period in the name succeeds.

Marc van Dongen provided the basic structure for this macro.

`\PName`

```

115 \DeclareDocumentCommand\PName{s o m o m}%
116 {%
117   \IfBooleanTF{#1}{\Name*[#2]{#3}}{\Name[#2]{#3}}%
118   {\space}\AKA[#2]{#3}[#4]{#5}}%
119 }
```

`\PName` is a convenience macro whose starred and unstarred forms call the respective versions of `\Name`, then `\AKA`. It prevents both the “sobriquet” feature and the alternate forenames feature.

\AKA

```

120 \DeclareDocumentCommand\AKA{o m o m}%
121 {%
122     \@nameauth@Punctfalse%
123     \IfValueTF{#3}%
124     {\@nameauth@FmtName*{#3 #4}%
125         \ifcsname\@nameauth@CleanName{#3#4!PN!}\endcsname\relax\else%
126             \csgdef{\@nameauth@CleanName{#3#4!PN!}}{}%
127             \IfValueTF{#1}%
128                 {\index{#4,{\space}#3|see{#2,{\space}#1}}}%
129                 {\index{#4,{\space}#3|see{#2}}}%
130             \fi}%
131     {\@nameauth@FmtName*{#4}%
132         \ifcsname\@nameauth@CleanName{#4!PN!}\endcsname\relax\else%
133             \csgdef{\@nameauth@CleanName{#4!PN!}}{}%
134             \IfValueTF{#1}%
135                 {\index{#4|see{#2,{\space}#1}}}%
136                 {\index{#4|see{#2}}}%
137             \fi}%
138     \if@nameauth@Punct\expandafter\@nameauth@CheckDot\fi%
139 }

```

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

\IndexName

```

140 \DeclareDocumentCommand\IndexName{o m o}%
141 {%
142     \IfValueTF{#1}%
143     {\ifcsname\@nameauth@CleanName{#1#2!PN!}\endcsname%
144         \else\index{#2,{\space}#1}\fi}%
145     {\IfValueTF{#3}%
146         {\ifcsname\@nameauth@CleanName{#2#3!PN!}\endcsname%
147             \else\index{#2 #3}\fi}%
148         {\ifcsname\@nameauth@CleanName{#2!PN!}\endcsname%
149             \else\index{#2}\fi}%
150     }%
151 }

```

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

\ForgetName

```

152 \DeclareDocumentCommand\ForgetName{o m o}%
153 {%
154     \IfValueTF{#1}%
155     {\csundef{\@nameauth@CleanName{#1#2!MN!}}}%
156     {\csundef{\@nameauth@CleanName{#1#2!NF!}}}%

```

```

157         {\IfValueTF{#3}%
158             {\csundef{\@nameauth@CleanName{#2#3!MN!}}}%
159             \csundef{\@nameauth@CleanName{#2#3!NF!}}}%
160         {\csundef{\@nameauth@CleanName{#2!MN!}}}%
161         \csundef{\@nameauth@CleanName{#2!NF!}}}%
162     }%
163 }

```

`\ForgetName` uses the ε -TeX facilities to undefine control sequences created by `\Name`. Using `\ifdefined` does not create a control sequence equivalent to `\relax`, thus using less resources.

`\NamesInactive`

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

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

`\NamesActive`

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

Change History

v0.7		
General: Initial version	1
v0.75		
General: New features described; documentation clarified	1
\ForgetName: New parameter added	18
\IndexName: Optional parameter added; mandatory parameter deleted	18
\Name: Added “sobriquet” feature		15
v0.8		
\nameauth@CheckDot: Renamed macro to help compatibility	..	14
\nameauth@CleanName: Renamed macro to help compatibility	..	14
\nameauth@EvalDot: Renamed macro to help compatibility	..	14
\nameauth@FmtName: Renamed macro to help compatibility	..	15
\nameauth@TestDot: Renamed macro to help compatibility	..	14
General: Added quick start guide for the impatient	3
Expanded description of func- tionality	12
Refactoring to improve function- ality and compatibility; docu- mentation expanded	1
\Name: Merged all major function- ality here	15

Index

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

Symbols	. 26, 37, 44, 45,	Arouet, François-Marie
\@@Zapsuffix 15, 16 <i>see</i> Voltaire
\@ForeNames	
. 35, 44, 48, 53, 57	79, 80, 83, 88,	B
\nameauth@CheckDot	95, 96, 99, 104,	Berenstain, Jan
..... 18, 113, 138	105, 108, 124, 131 8
\nameauth@CleanName		Berenstain, Stan
..... 13, 36,	\nameauth@Punctfalse 8
42, 47, 51, 56, 23, 33, 122	Boethius
62, 68, 73, 77,	\nameauth@Puncttrue 5
82, 87, 93, 98, 22	C
102, 107, 125,	\nameauth@TestDot	Carnap, Rudolph
126, 132, 133, 20, 28 13
143, 146, 148,	\nameauth@Zapsuffix	Charles the Bald
155, 156, 158–161 14, 45, 5
\nameauth@DoFormatfalse	54, 71, 80, 96, 105	Chiang Kai-shek
..... 4, 164	\@period 7
\nameauth@DoFormattrue 19	Cicero, M.T.
..... 3, 165	\@token 5, 7
\nameauth@EvalDot 18, 19	Clemens, Samuel L.
..... 18, 19		... <i>see</i> Twain, Mark
\nameauth@FmtName		Confucius
	A 5
	Adams, John Q.	\csgdef
 6, 7	.. 47, 56, 73,
	\AKA	82, 98, 107, 126, 133
 10, 118, 120	\csundef
	Antiochus
	IV	. 155, 156, 158–161
	Epiphanes	
	.. 6, 8	D
	De Wette, Wilhelm	

M.L.	7	128, 129, 135,		R	
\detokenize	13	136, 144, 147, 149		Rambam	10,
<i>Doctor Angelicus</i>	<i>see</i>	\IndexName	8, <u>140</u>		<i>see also</i> Maimonides
Thomas Aquinas					
Dongen, Marc van . .	1, 17	J		S	
E		Jean sans Peur	9, 12	Snel van Royen,	
Einstein, Albert	5	Jean the Fearless . . .		Rudolph	11
Eliot, George	10	<i>see</i> Jean sans Peur		Snel van Royen, Wille-	
Evans, Mary Anne . .		K		brord	10, 11
. <i>see</i> Eliot, George		King, Jr., Martin		Snellius <i>see</i> Snel van	
\expandafter		Luther	6	Royen, Wille-	
. 13, 19, 113, 138				brord, <i>see</i> Snel	
F				van Royen, Rudolph	
\ForgetName	13, <u>152</u>	L		Stephani, Philipp	1
\futurelet	18	Lueck, Uwe	1, 14	Sun Yat-sen	7
G		M		SurnameA, Suffix, Fore-	
Goethe, Johann Wolf-		Maimonides	10	name	7
gang von	5	Malebranche, Nicolas	13	SurnameB, Suffix, Fore-	
Gregorio, Enrico	1	Mencius	6	name	7
		Moses ben-Maimon . .		T	
	 <i>see</i> Maimonides		\TestDot	21, 24
H		N		\TestEnd	21, 24
Hammerstein, II, Oskar	6	\Name	5, <u>31</u> , 117	\TestPunct	21, 22
Henry VIII	6	name forms	5	\TestStop	21, 24
I		\Name*	5	Thomas Aquinas	9, 12
\if@nameauth@DoFormat		\NamesActive	13, <u>165</u>	Thomas of Aquino <i>see</i>	
. 1, 41, 67, 92		\NamesFormat	5–8, 12, 29	Thomas Aquinas	
\if@nameauth@Punct .		\NamesInactive	<u>164</u>	Truman, Harry S.	6
. 2, 113, 138		\NamesInctive	13	Twain, Mark	10
\ifcsname	36, 42,	O		V	
51, 62, 68, 77,		Oberdiek, Heiko . . .	1, 14	Vlad II Dracul	12
87, 93, 102, 125,		P		Vlad III Dracula	12
132, 143, 146, 148		\PackageWarning . . .		Voltaire	10
\index	44, 45, 48, 38, 64, 89		W	
53, 54, 57, 70,		\PName	9, <u>115</u>	Walther, C.F.W.	8
71, 74, 79, 80,		Ptolemy I Soter	6	Z	
83, 95, 96, 99,				\zap@space	13
104, 105, 108,					