Genealogical Profiles for LaTeX

Mikkel Eide Eriksen
mikkel.eriksen@gmail.com

December 16, 2025

Preface

This package enables the presentation of individual *profiles*, which may be useful for genealogical or local history treatises.

Each profile is typeset using key/value-configurable environments, and a number of macros are provided to enable formatting, references, and floruit (lifespan) calculations.

In order to provide an ergonomic interface for document authors, extra care has been taken to enable automatic and configurable parsing of names.

Issues can be reported at
https://github.com/mikkelee/latex-genprofile/issues

Wolfgang Amadeus MOZART **WM1**

★ January 27, 1756 in Getreidegasse 9, Salzburg, † December 5, 1791 in Vienna.

Had a complex relationship with his rival Antonio SALIERI$^{\text{AS1}}_{\text{p. 1}}$.

Antonio SALIERI **AS1**

★ August 18, 1750 in Legnago, Republic of Venice, † May 7, 1825 in Vienna.

Was falsely accused of poisoning MOZART$^{\text{WM1}}_{\text{p. 1}}$.

1 Configuration

The package is configured in the following manner:

```
\usepackage{genealogy-profiles}
```

Loads the package and sets some sensible defaults as further described in sections 5.1 and 5.2.

```
\gprKeys{⟨general options⟩}
```

Can be used to set options globally (in the preamble) or locally (in a group). See section 5.1 for possible keys/values.

2 Usage

2.1 Environments

```
\begin{gprProfile}[⟨profile options⟩]{⟨full name⟩}[⟨life events⟩]
  ⟨environment content⟩
\end{gprProfile}
```

Typesets its contents according to the configured layout (see section 2.3). For possible profile options, see section 5.2.
The ⟨*full name*⟩ will be parsed according to the current `name part order`→P. 12 (see section 3 for discussion) and styled according to ⟨*name part*⟩ `style`→P. 10 etc.
Each profile must have an ⟨*id*⟩, either supplied by the author via the `id`→P. 13 profile option key or automatically generated via the general `auto id`→P. 8 key (the latter being the default).
The `life events`→P. 15 will be parsed as a database by the `genealogytree`→CTAN package.

```
\begin{gprProfile*}{⟨profile options⟩}[⟨life events⟩]
  ⟨environment content⟩
\end{gprProfile*}
```

The `gprProfile*` environment provides more control at the expense of convenience, by requiring the name part(s) be set explicitly as profile options. See `given name`→P. 15 etc.

2.2 Commands

`\gprName{⟨name⟩}`

`\gprName*{⟨name⟩}`

Typesets a name, styled according to configured name styles (see ⟨*name part*⟩ `style`→ P. 10), without attempting to link to a profile.
The regular version adds the name to any configured indices, the starred version does not.

`\gprRef[⟨id⟩]{⟨name⟩}`

`\gprRef*[⟨id⟩]{⟨name⟩}`

Typesets a reference to a profile according to `reference style`→ P. 13.
The ⟨*id*⟩ is optional, in case it is not known or available at the time of writing. However, if the name is unique to the document, the reference should automatically be recognized.
If it is not possible for the package to identify the intended reference, either via ⟨*id*⟩ or unique name, warnings will be emitted in the log as well as in the document (the latter can be configured via `unknown reference style`→ P. 13).
When using an ⟨*id*⟩, the name parameter can be left empty, or alternatively be used to override the by-default displayed full name, eg. to change case to genitive, etc.
The regular version adds the reference to configured indices, the starred version does not.
When used inside a profile, the package will emit a warning if the referenced profile does not refer back to the originating profile, in order to allow the author to check referential integrity (if they so wish).
These commands require two runs to account for forward references.

`\gprLifeEvents{⟨id⟩}`

`\gprLifeEvents*{⟨od⟩}`

Typesets the stored `life events`→ P. 15 for a given ⟨*id*⟩.
Requires `life events storage`→ P. 12 to be either `memory` or `file`.

`\gprTreeContentInterpreter`

This command is used with the `content interpreter` key of `genealogytree`→CTAN.
When used, it handles tree node content in the follow way:

- If the node content is wrapped in `{`⟨*brackets*⟩`}`, it is interpreted as a profile ⟨*id*⟩, and the name and life events are passed to `genealogytree`→CTAN to typeset.
- If the node content is not wrapped, it is interpreted as a regular `genealogytree`→CTAN database, and will be typeset per its settings.

See example in section 6.3.

2.3 Layout

The typeset profiles are laid out according to the following structure. See also `layout preset`→P. 11.

1. The contents of `begin profile`→P. 9
2. If `auto header`→P. 8 is `true`:
    (a) The contents of `begin header`→P. 9
    (b) The contents of `header format`→P. 9
    (c) The contents of `end header`→P. 9
3. If `include life events`→P. 11 is `before content`, and `life events`→P. 15 and/or `\gprYear`→P. 8(s) were used:
    (a) The contents of `begin life events`→P. 9.
    (b) The events formatted according to the database format configured via the `genealogytree`→CTAN package.
    (c) The contents of `end life events`→P. 9
4. The contents of `begin content`→P. 9
5. The content provided to the environment by the author.
6. The contents of `end content`→P. 9
7. If `include life events`→P. 11 is `after content`, and `life events`→P. 15 and/or `\gprYear`→P. 8(s) were used:
    (a) The contents of `begin life events`→P. 9.
    (b) The events formatted according to the database format configured via the `genealogytree`→CTAN package.
    (c) The contents of `end life events`→P. 9
8. The contents of `end profile`→P. 9

3 Names

Names are parsed according to the configured `name part order`$^{\rightarrow\text{P. 12}}$ (some presets are provided via `name type`$^{\rightarrow\text{P. 12}}$), in a left-to-right evaluation.

In order for single name parts to include multiple separate tokens (eg. multiple given names), underscores can be used to combine them. For example, `Wolfgang_Amadeus Mozart` will, with the default `given and surname` name type, be parsed as the given name(s) `Wolfgang Amadeus` and the surname `Mozart`.

If the `name part order`$^{\rightarrow\text{P. 12}}$ has more parts than the supplied value, the right-most parts will be empty. If this is not desired, one may mark empty name parts with a single underscore; for example, using the `nordic historical` name type, `Jens _ Smed` will parse as the given name `Jens`, no patronymic, and the byname `Smed` (ie. blacksmith).

Inside a profile environment, a number of shortcuts are provided to access the available name parts, as well as the ⟨*id*⟩ and a full name styled according to the individual name part style keys (see section 4).

3.1 Creating and Using Name Parts

`\gprCreateNamePart{⟨name part⟩}[⟨macro name⟩]`

If the supplied default name part types are not sufficient, new types can be created. They must be included in the `name part order`$^{\to \text{P. } 12}$ to be properly recognized during name parsing and indexing.
Creating a ⟨*name part*⟩ results in:

- ⟨*name part*⟩$^{\to \text{P. } 15}$ key in profile options for manually setting the name part in profiles.
- ⟨*name part*⟩ `style`$^{\to \text{P. } 10}$ key in general options for styling the name part.
- ⟨*name part*⟩ `index`$^{\to \text{P. } 10}$ key in general options for which index to use for the name part.
- `\gpr`⟨*NamePart*⟩$^{\to \text{P. } 7}$ macro for typesetting the styled name part within a profile.
- `\gpr`⟨*NamePart*⟩`*`$^{\to \text{P. } 7}$ macro for typesetting the unstyled name part within a profile.
- `\gpr`⟨*NamePart*⟩`Style`$^{\to \text{P. } 7}$ macro for typesetting arbitrary text in the configured name part style.

By default, the generated commands will be CamelCased (eg. `given name` results in `\gprGivenName...`), but this can be overridden by supplying your own base name as the optional argument.
The name parts `given name`, `patronymic`, `surname`, and `byname` have been pregenerated, along with their associated style/index keys and commands (see sections 4 and 5.2).

4 Profile Macros

A number of extra macros are available inside profiles to allow accessing some key values, as well as gather years for floruit calculations.

`\gprHeader{}`

A header styled according to `header format`$^{\to \text{P. } 9}$.
If `auto header`$^{\to \text{P. } 8}$ is `false`, one may wish to use `\gprHeader` to manually insert the header at the desired location (see section 2.3 for layout structure and section 6 for an example).

```
\gprID{}
\gprID*{}
\gprFullName{}
\gprFullName*{}
```

Typesets the ⟨*id*⟩ and full name according to configured styles (see `id style`→P. 10 and ⟨*name part*⟩ `style`→P. 10). The starred versions of the commands are unstyled.

```
\gprGivenName{}
\gprGivenName*{}
\gprPatronymic{}
\gprPatronymic*{}
\gprSurname{}
\gprSurname*{}
\gprByname{}
\gprByname*{}
\gpr⟨NamePart⟩{}
\gpr⟨NamePart⟩*{}
```

Typesets individual name parts according to configured styles (see ⟨*name part*⟩ `style`→P. 10). The starred versions of the commands are unstyled.
More can be created using `\gprCreateNamePart`→P. 6.

```
\gprIDStyle{⟨text⟩}
\gprGivenNameStyle{⟨text⟩}
\gprPatronymicStyle{⟨text⟩}
\gprSurnameStyle{⟨text⟩}
\gprBynameStyle{⟨text⟩}
\gpr⟨NamePart⟩Style{⟨text⟩}
```

Styles given text according to the configured name style (see ⟨*name part*⟩ `style`→P. 10).
More can be created using `\gprCreateNamePart`→P. 6.

`\gprYear{⟨year⟩}`

`\gprYear*{⟨year⟩}`

All tagged years in a profile will be gathered and inserted as a `floruit` range under `life events`→P. 15, which will by default only be displayed if there is no defined lifespan (ie. birth or baptism *and* death or burial).
The starred version does not typeset anything, and can thus be used to add “hidden” years to the floruit event.
These commands require two runs.

`\gprYears{⟨year range⟩}`

`\gprYears*{⟨year range⟩}`

Adds two years to the `floruit` event, by splitting at one or more hyphens.
Values such as `1750--1755` or `1750--55` will both be parsed as the two years `1750` and `1755` and typeset as the expected 1750–1755 or 1750–55, respectively.
Like `\gprYear`, the starred version produces no output, and two runs are required.

5 Option Keys

5.1 General Options

These are used with the `\gprKeys`→P. 2 command, either globally in the preamble or locally in a group.

`auto header=⟨true/false⟩` (initially `false`)

Automatically inserts a header using `header format`→P. 9 at the beginning of profiles. See section 2.3.

`auto id=⟨true/false⟩` (initially `true`)

Automatically generates an ⟨*id*⟩ for each profile if no `id`→P. 13 is supplied. They are built from the initials of each the name part (empty parts replaced by a dash) combined with a number to ensure uniqueness.
If a profile has no ⟨*id*⟩ set, whether automatically or manually, an error is emitted.

`auto id prefix`=⟨...⟩ (initially not set)

Prefixes auto-generated ⟨*id*⟩s with a string, which may be useful for works containing multiple logically or temporally separate chapters or sections.

`begin profile`=⟨...⟩ (initially not set)
`begin header`=⟨...⟩ (initially not set)
`end header`=⟨...⟩ (initially not set)
`begin life events`=⟨...⟩ (initially not set)
`end life events`=⟨...⟩ (initially not set)
`begin content`=⟨...⟩ (initially not set)
`end content`=⟨...⟩ (initially not set)
`end profile`=⟨...⟩ (initially not set)

These keys allow configuring arbitrary LaTeX code to be inserted before/during/after the typeset `gprProfile`→ P. 2 and `gprProfile*`→ P. 2 environments (see section 2.3 for layout structure).

`clear name styles`=⟨*true/false*⟩ (initially not set)

Clears all ⟨*name part*⟩ `style`→ P. 10 keys in the current `name part order`→ P. 12.

`header format`=⟨...⟩ (initially `{\gprFullName{}\hfill\gprID{}}`)

Formats a profile header, which will be available as `\gprHeader`→ P. 6, and is automatically inserted if `auto header`→ P. 8 is `true`. See section 4 for available macros.

`id index`=⟨...⟩ (initially not set)
`full name index`=⟨...⟩ (initially not set)
`given name index`=⟨...⟩ (initially not set)
`patronymic index`=⟨...⟩ (initially not set)
`surname index`=⟨...⟩ (initially not set)
`byname index`=⟨...⟩ (initially not set)
⟨*name part*⟩ `index`=⟨...⟩ (initially not set)

Setting these keys will cause usage of `gprProfile`→P. 2, `\gprName`→P. 3, and `\gprRef`→P. 3 to emit index data to the named index unless disabled using `no index`→P. 15. The index must be created with eg. `imakeidx` before usage.
Multiple keys can point to the same index. For example, you can use the same index for patronymics, surnames, and/or bynames — which has in fact been configured for this document (see final page).
At the end of the document, `\printindex` can then be used for each configured index as normal.
More can be created using `\gprCreateNamePart`→P. 6.

`id style`=⟨...⟩ (initially not set)
`given name style`=⟨...⟩ (initially not set)
`patronymic style`=⟨...⟩ (initially not set)
`surname style`=⟨...⟩ (initially not set)
`byname style`=⟨...⟩ (initially not set)
⟨*name part*⟩ `style`=⟨...⟩ (initially not set)

These keys set the styling for the ⟨*id*⟩ and name parts, which will be used when typesetting the name parts with the `\gpr`⟨*NamePart*⟩→P. 7 commands, and can also be applied to arbitrary strings using the `\gpr`⟨*NamePart*⟩`Style`→P. 7 commands.
The styled name parts will together be available as the `\gprFullName`→P. 7.
More can be created using `\gprCreateNamePart`→P. 6.

`id in index entries`=⟨*true/false*⟩ (initially not set)

Causes index entries to include the ⟨*id*⟩ in parentheses, which is especially useful in documents with profiles for several identically named persons.

`include unknown in index`=⟨*true/false*⟩ (initially not set)
`include ambiguous in index`=⟨*true/false*⟩ (initially not set)

Causes the indexes to include mentions (`\gprRef`→P. 3 and `\gprName`→P. 3) of persons with unknown ⟨*id*⟩s or ambiguous names, which can be used for correcting drafts.

`include life events`=⟨*...*⟩ (initially `before content`)

- `false`: Life events will not be automatically included in the typeset profiles.
- `before content`: Life events will be typeset before the main content of the profile (the default).
- `after content`: Life events will be typeset after the main content of the profile.

See section 2.3.

`layout preset`=⟨*...*⟩ (initially not set)

Sets a preconfigured layout style. Currently, two options are available:

- `clear` will blank out all the layout keys (see `begin profile`→P. 9 etc).
- `tcolorbox` sets the following keys:

```
\gprKeys{
  auto header = false,
  begin profile =
  ↪  {\begin{tcolorbox}[title=\gprHeader]},
  end life events = {\tcblower},
  end profile = {\end{tcolorbox}}
}
```

  The box can be further configured via `\tcbset` as per the `tcolorbox`→CTAN documentation.

Suggestions for more presets are welcome.

`life events storage`=⟨...⟩ (initially `none`)

Configures stored life events, enabling the use of `\gprLifeEvents`→P. 3 and `\gprTreeContentInterpreter`→P. 4.

- `none`: Life events will not be stored (the default).
- `memory`: Life events will be stored in memory, meaning they can only be accessed after the profile has been typeset.
- `file`: Life events will be cached in the .aux file, enabling use anywhere in a document (requires two runs to ensure full sync).

`main index entry style`=⟨...⟩ (initially not set)

Adds formatting to the "main" index entry page numbers (ie. the ones pointing to the actual profile), leaving the ones that are merely referenced with `\gprRef`→P. 3 untouched; for example `textbf` will bold the main entry.

`name part order`=⟨...⟩ (initially `{given name, surname}`)
`name type`=⟨...⟩ (initially `given and surname`)

The `name part order` is used for splitting a `full name`→P. 14 to its constituent parts for the configured styling, indexing, and autogenerated ⟨*id*⟩s (see section 3).
New name parts must be created with `\gprCreateNamePart`→P. 6 before inclusion in the name part order.
Using the `name type` key provides access to a number of preconfigured `name part orders`:

- `given and surname` will set the `name part order` key to `{given name, surname}` (the default).
- `nordic historical` will set the `name part order` key to `{given name, patronymic, byname}`, which was traditionally used in Scandinavia and the rest of the nordic countries.

If no preset `name type` exists for the intended use case, the `name part order` can be set directly (suggestions for other `name types` are welcome).

`nest index entries`=⟨*true/false*⟩ (initially not set)

Causes index entries to be nested under the various patronymics/-surnames/bynames.

`page reference style`=⟨*...*⟩ (initially `p.\nobreakspace#1`)

Formats page references. The value is expanded with the argument `#1` being the page number.

`reference style`=⟨*...*⟩ (see below)

Formats references to known profiles. The value is expanded with the arguments `#1` being the name, `#2` being the ⟨*id*⟩, and `#3` being the formatted page reference as per `page reference style`.
The default is to present these as the name followed by combined super- and subscripts (see page 1 for the default, also section 6 for another style).
The result is wrapped in a `hyperref`→CTAN link to the referenced profile.

`unknown reference style`=⟨*...*⟩ (see below)

Formats references to unknown profiles. The value is expanded with the arguments `#1` being the supplied name, and `#2` being a short description of the reason.
The default is to present the name as red text with the reason (unknown/ambiguous) following in parentheses.
Note that forward references will appear unknown on the first run.

`use styles in index`=⟨*true/false*⟩ (initially not set)

Causes index entries to be styled according to the various ⟨*name part*⟩ `style`→P. 10 keys.
The index at the end of this file shows it in use.

5.2 Profile Options

These are used in the first argument of the `gprProfile`→P. 2 and `gprProfile*`→P. 2 environments.

`id`=⟨*...*⟩ (initially not set)

Sets an ⟨*id*⟩ for the profile. If none is specified and `auto id`→P. 8 is `true`, one will be generated from name initials combined with a number to ensure uniqueness.
Not specifying an ⟨*id*⟩ while `auto id`→P. 8 is `false` will cause an error.
Likewise, if an already used ⟨*id*⟩ is specified, an error will be emitted.

`full name={⟨...⟩}` (initially not set)

Sets the full name of the person. If it is specified, the configured `name part order`$^{\to\text{P. }12}$ will be used to determine the individual name parts.

If it is not specified, one will be generated by combining the given name parts according to the configured `name part order`$^{\to\text{P. }12}$, using the below keys. See section 3 for further details.

This key is set by the mandatory argument of `gprProfile`$^{\to\text{P. }2}$.

In the starred version of the environment, either the `full name` or individual ⟨*name part*⟩$^{\to\text{P. }15}$ keys must be set.

`given name`=⟨...⟩ (initially not set)
`patronymic`=⟨...⟩ (initially not set)
`surname`=⟨...⟩ (initially not set)
`byname`=⟨...⟩ (initially not set)
⟨*name part*⟩=⟨...⟩ (initially not set)

Sets individual name parts, to be used with `gprProfile*`→P. 2, if one prefers not to use the auto-parsing of `gprProfile`→P. 2.
More name parts can be created using `\gprCreateNamePart`→P. 6.

`life events`=⟨...⟩ (initially not set)

Populates a `genealogytree`→CTAN database, which will be typeset using `\gtrPrintDatabase`, according to the settings of that package; refer to its documentation for configuration.
The default provided by `genealogy-profiles` simply lists life events separated by commas, skipping `floruit` if there is a defined lifespan (ie. birth or baptism *and* death or burial). Note that marriage is not included in the default, but can be enabled by changing the `genealogytree`→CTAN `database format` key.
For convenience, this key can be set with the final optional argument of the `gprProfile`→P. 2 and `gprProfile*`→P. 2 environments.
See also `life events storage`→P. 12, `\gprLifeEvents`→P. 3, and `\gprTreeContentInterpreter`→P. 4.

`no index`=⟨*true/false*⟩ (initially not set)

If set, no index entries will be emitted for this profile.

6 Examples

6.1 Using `tcolorbox` & `nordic historical`

A simple example to show name parsing and use of `\gprRef`$^{\to\text{P. 3}}$ and `\gprYear`$^{\to\text{P. 8}}$.

```
\gprKeys{
  name type = nordic historical,
  patronymic style = \itshape,
  byname style = \scshape,
  layout preset = tcolorbox,
}

\begin{gprProfile}{Jens Hansen}[ birth = {1790}{Denmark} ]
  Wife: \gprRef{Anne_Marie Olsdatter}.

  Let's also tag some years:
  \gprName{Jens} is mentioned in the censuses of
  \gprYear{1801}, \gprYear{1834}, and \gprYear{1840}.
\end{gprProfile}

\begin{gprProfile}{Anne_Marie Olsdatter}[ birth = {1795}{Denmark} ]
  Husband: \gprRef{Jens Hansen}.
\end{gprProfile}

\begin{gprProfile}{Jens Hansen Smed}
  An unrelated person with a byname.

  Not to be confused with \gprName{Jens _ Smed} of Neighbouring Town.
\end{gprProfile}
```

Jens *Hansen* JH-1

★ 1790 in Denmark, ⁘ 1801 to 1840.

Wife: Anne Marie *Olsdatter*$^{\text{AO-1}}_{\text{p. 16}}$.
Let’s also tag some years: Jens is mentioned in the censuses of 1801, 1834, and 1840.

Anne Marie *Olsdatter* AO-1

★ 1795 in Denmark.

Husband: Jens *Hansen*$^{\text{JH-1}}_{\text{p. 16}}$.

Jens *Hansen* SMED JHS1

An unrelated person with a byname.
Not to be confused with Jens SMED of Neighbouring Town.

6.2 Using Custom Layout & `given and surname`

A simple example to show different layout and reference styles.

```
\newcommand\spacedrule{\vspace*{5pt}\hrule\vspace*{5pt}}
\gprKeys{
  name type = given and surname,
  surname style = \scshape,
  auto header,
  begin profile = \spacedrule,
  end life events = \spacedrule,
  end profile = {\spacedrule\vspace*{\baselineskip}},
  reference style = {#1\footnote{#2,~#3}}
}

\begin{gprProfile}{George Washington}[
  birth = {1732-02-22}{Popes Creek, Virginia Colony},
  death = {1799-12-14}{Mount Vernon, Virginia, U.S.}
]
  Attended the first \gprRef[WM1]{_ Mozart} performance
  in America in \gprYear{1784}.
\end{gprProfile}

This sentence demonstrates that references work even outside
profiles: \gprRef[GW1]{_ Washington} was the first president
of the United States.
```

George Washington GW1★ February 22, 1732 in Popes Creek, Virginia Colony, ✝ December 14, 1799 in Mount Vernon, Virginia, U.S..

Attended the first Mozart[a] performance in America in 1784.

This sentence demonstrates that references work even outside profiles: Washington[b] was the first president of the United States.

---

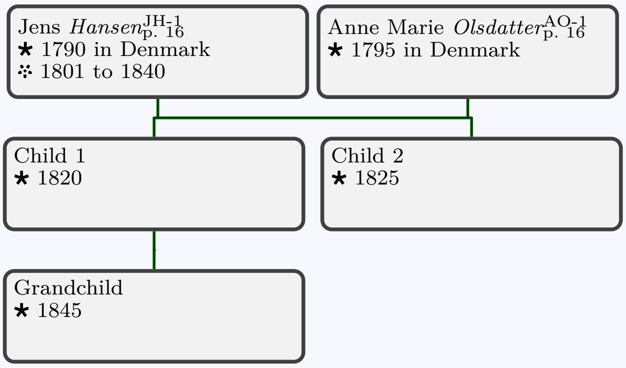
[a] WM1, p. 1

[b] GW1, p. 17

6.3 Using `\gprTreeContentInterpreter`→ P. 4

A simple example to show the use of stored life events in a `genealogytree`→ CTAN. This requires setting the `life events storage`→ P. 12 key.

```
\gprKeys{
  name type = nordic historical,
}
\begin{tikzpicture}
\genealogytree[
  content interpreter=\gprTreeContentInterpreter,
  database format=full,
  node size=4cm,
  level size=1.25cm
]{
  child{
    g{{JH-1}}
    p{{AO-1}}
    child{
      g{ name = {Child 1}, birth- = {1820} }
      c{ name = {Grandchild}, birth- = {1845} }
    }
    c{ name = {Child 2}, birth- = {1825} }
  }
}
\end{tikzpicture}
```



Index

Profile Index

This index was created by setting the keys `patronymic index`, `surname index`, and `byname index` to identical values. Notice that Jens *Hansen* SMED is listed both under his patronymic *Hansen* and his byname SMED.