

The nehgs-register package*

James P. Howard, II
jh@jameshoward.us

2026/08/10

Abstract

The `nehgs-register` package provides commands and environments for typesetting genealogical registers in the style used by the *New England Historical and Genealogical Register* (NEHGS). It manages a person registry with computed generation numbers, ancestor chains, child lists with continuation markers, multiple marriages, vital records, date formatting, gender-aware pronouns, source citations, index generation, and cross-references.

Contents

1	Introduction	2
2	Quick start	2
3	Package options	4
4	Defining persons	4
5	Marriage support	5
6	Name retrieval	6
7	Vital record accessors	6
8	Generation computation	6
9	Register entries	6
10	Child lists	7
11	Approximate dates	7
12	Gender-aware pronouns	7
13	Source citations	7
14	Date formatting	8

*This document corresponds to `nehgs-register` v1.1, dated 2026/08/10.

15 Shorthands	8
16 Register section	8
17 Validation	8
18 Implementation	8
18.1 Package identification and options	8
18.2 Required packages	10
18.3 Internal helpers	10
18.4 Person registry	10
18.5 Multiple marriage support	13
18.6 Name retrieval	14
18.7 Vital record accessors	15
18.8 Gender-aware pronouns	16
18.9 Approximate dates	17
18.10 Generation computation	17
18.11 Ancestor chain formatting	18
18.12 Main register entries	20
18.13 Cross-references	21
18.14 Child lists	21
18.15 Source and note support	22
18.16 Automatic generation headings	22
18.17 Date formatting	23
18.18 Date validation	23
18.19 Validation commands	23
18.20 Shorthands	25
18.21 Register section environment	25
18.22 Backward compatibility	25

1 Introduction

The Register style, also called the NEHGS style, is the standard format for presenting genealogical data in descending order. Each person who continues the lineage receives a numbered main entry; children are listed under their parents with roman numeral labels. Children who appear later as main entries are marked with a “+” continuation symbol.

This package automates the mechanical parts of Register typesetting: numbering, generation computation, ancestor chains, parent validation, and index entries. The genealogical prose itself remains the author’s responsibility.

2 Quick start

```
\documentclass{article}
\usepackage[en-US]{datetime2}
\usepackage[date=en-US,abbrevparents]{nehgs-register}

\definepersonkv{john-doe}{
  name      = {John Doe},
```

```

    preferred    = {John},
    parents      = {},
    sex          = {M},
    birth        = {1820-03-15},
    birth-place  = {Boston, Massachusetts}
}

\definepersonkv{mary-smith}{
    name         = {Mary Smith},
    preferred    = {Mary},
    parents      = {},
    sex          = {F},
    birth        = {1825-06-20}
}

\definepersonkv{james-doe}{
    name         = {James Doe},
    preferred    = {James},
    parents      = {john-doe,mary-smith},
    lineage-parent = {john-doe},
    sex          = {M},
    birth        = {1850-11-02}
}

\addmarriage{john-doe}{
    spouse = mary-smith,
    date   = 1845-09-10,
    place  = {Springfield, Massachusetts}
}

\begin{document}
\begin{registersection}

\generationheading{john-doe}
\mainperson{john-doe} was born
\getpersonbirthdate{john-doe} in
\getpersonbirthplace{john-doe}.
\HeSheThey{john-doe} married
\getpersonname{mary-smith} on
\getmarriagedate{john-doe}{1} in
\getmarriageplace{john-doe}{1}.

Children of John Doe and Mary Smith:
\begin{childrenof}{john-doe}{mary-smith}
    \childref{james-doe}{b.~\getpersonbirthdate{james-doe}.}
\end{childrenof}

\generationheading{james-doe}
\mainperson{james-doe} was born
\getpersonbirthdate{james-doe}.

\end{registersection}
\checkregister
\end{document}

```

3 Package options

The package accepts the following options:

index (default)

Load `imakeidx` and generate index entries whenever a person's name is printed by `\getpersonname`.

noindex

Suppress all index generation.

fullparents (default)

Print full names in parenthetical ancestor chains.

abbrevparents

Print preferred (short) names in ancestor chains.

ngsq

Use NGSQ (Modified Register) numbering.

datestyle=*style*

Set the `datetime2` date style at `\begin{document}`. The synonym `date=` is also accepted.

numbering=ngsq

Equivalent to the `ngsq` option.

4 Defining persons

`\definepersonkv \definepersonkv{<id>}{<key=value pairs>}`

The primary interface for declaring a person in the registry. The `<id>` is a short identifier (e.g., `john-doe`) used to refer to this person throughout the document. Available keys:

name

Full name as it should appear in the register.

preferred

Short or preferred name for abbreviated contexts.

parents

Comma-separated list of parent ids. Leave empty for the founding generation.

index

Index entry text. Defaults to the `name` value.

lineage-parent

Single parent id for lineage chain display. When set, ancestor chains follow only this parent rather than listing all parents.

sex M, F, or empty. Used by pronoun helpers.

birth
Birth date in ISO format (YYYY-MM-DD).

birth-place
Birth place as free text.

death
Death date in ISO format.

death-place
Death place as free text.

baptism
Baptism or christening date in ISO format.

baptism-place
Baptism or christening place.

burial
Burial date in ISO format.

burial-place
Burial place.

occupation
Occupation as free text.

marriage-date
Date of (first) marriage in ISO format. For multiple marriages, use `\addmarriage` instead.

marriage-place
Place of (first) marriage.

`\defineperson` `\defineperson{<id>}{<parents>}{<name>}{<preferred>}` [`<index>`]
Legacy positional interface. Prefer `\definepersonkv` for new work.

5 Marriage support

`\addmarriage` `\addmarriage{<person-id>}{<key=value pairs>}`
Add a marriage record to a previously defined person. Supports multiple marriages per person. Available keys:

date
Marriage date in ISO format.

place
Marriage place.

spouse
Person id of the spouse (must be separately defined).

Marriages are numbered sequentially starting from 1 in the order they are added. If `\marriage-date` or `\marriage-place` is given inline in `\definepersonkv`, that counts as marriage 1.

<code>\getmarriagedate</code>	<code>\getmarriagedate{<person-id>}{<number>}</code> Print the date of marriage <code><number></code> , formatted through <code>\datetime2</code> .
<code>\getmarriageplace</code>	<code>\getmarriageplace{<person-id>}{<number>}</code> Print the place of marriage <code><number></code> .
<code>\getmarriagespouse</code>	<code>\getmarriagespouse{<person-id>}{<number>}</code> Print the name of the spouse from marriage <code><number></code> (via <code>\getpersonname</code> , including index entry).
<code>\getmarriagecount</code>	<code>\getmarriagecount{<person-id>}</code> Expand to the number of recorded marriages for the person.

6 Name retrieval

<code>\getpersonname</code>	<code>\getpersonname{<id>}</code> prints the full name in small caps and generates an index entry.
<code>\getpersonnameNI</code>	<code>\getpersonnameNI{<id>}</code> prints the full name in small caps without an index entry.
<code>\getpersonpreferred</code>	<code>\getpersonpreferred{<id>}</code> prints the preferred (short) name in small caps.
<code>\getpersonabbrevNI</code>	<code>\getpersonabbrevNI{<id>}</code> prints the preferred name if set, otherwise the full name, without an index entry.
<code>\shortname</code>	<code>\shortname{<id>}</code> is an alias for <code>\getpersonpreferred</code> .

7 Vital record accessors

Each accessor prints the stored value, or a bracketed placeholder with a package warning if the value is missing.

<code>\getpersonbirthdate</code>	<code>\getpersonbirthdate{<id>}</code> , <code>\getpersonbirthplace{<id>}</code> , etc.
<code>\getpersonbirthplace</code>	
<code>\getpersondeathdate</code>	Date accessors format the stored ISO date through <code>\registerdate</code> . Place
<code>\getpersondeathplace</code>	and occupation accessors print the stored text verbatim. <code>\getpersonsex{<id>}</code>
<code>\getpersonbaptismdate</code>	expands to the raw stored value (M, F, or empty).
<code>\getpersonbaptismplace</code>	

8 Generation computation

<code>\getpersonburialdate</code>	
<code>\getpersonburialplace</code>	
<code>\getpersonoccupation</code>	<code>\getgeneration{<id>}</code> computes and prints the generation number. Persons with
<code>\getpersonsex</code>	no parents are generation 1; others are one more than their highest-generation
<code>\getgeneration</code>	parent. Results are cached after the first computation. Cycles are detected and
	produce a warning.
<code>\generationheading</code>	<code>\generationheading{<id>}</code> prints a bold heading such as “Third Generation”
	based on the person’s computed generation. Supports generations 1–20 by name;
	higher generations use the numeric form.

9 Register entries

<code>\mainperson</code>	<code>\mainperson{<id>}</code> typesets a numbered main register entry: the entry number
--------------------------	--

in bold, the person’s name in small caps with a generation superscript, and the parenthetical ancestor chain. Each call increments the `registerperson` counter and creates a label `person:⟨id⟩` for cross-references.

- `\xrefperson` `\xrefperson{⟨id⟩}` prints a hyperlinked cross-reference to a person’s main entry, showing their name and entry number.
- `\childname` `\childname{⟨id⟩}` prints a person’s name with generation superscript, without an index entry. Used in child lists.
- `\formatparents` `\formatparents{⟨id⟩}` prints the parenthetical ancestor chain. Called automatically by `\mainperson` but available for standalone use.

10 Child lists

- `children (env.)` `children` is an enumerated list with roman numeral labels, accepting optional `enumitem` keys.
- `childrenof (env.)` `childrenof{⟨parent-a⟩}{⟨parent-b⟩}[⟨enumitem keys⟩]` is like `children` but validates that each `\child` entry has the expected parent pair.
- `\child` `\child{⟨id⟩}{⟨description⟩}` typesets a child entry. The description text follows a comma after the child’s name.
- `\childref` `\childref{⟨id⟩}{⟨description⟩}` typesets a child entry with a “+” continuation marker, indicating that this child will appear later as a `\mainperson` entry. This is a core convention of the Register style.

11 Approximate dates

Genealogists frequently work with uncertain dates. These commands format common qualifiers used in genealogical writing:

- `\circa` `\circa{⟨date⟩}` prints “c. *⟨date⟩*”.
- `\before` `\before{⟨date⟩}` prints “bef. *⟨date⟩*”.
- `\after` `\after{⟨date⟩}` prints “aft. *⟨date⟩*”.
- `\between` `\between{⟨date1⟩}{⟨date2⟩}` prints “bet. *⟨date1⟩* and *⟨date2⟩*”.
- `\aboutdate` `\aboutdate{⟨date⟩}` prints “abt. *⟨date⟩*”.
- `\probably` `\probably{⟨date⟩}` prints “prob. *⟨date⟩*”.
- `\say` `\say{⟨date⟩}` prints “say *⟨date⟩*”.

These may be used both in prose and inside key values.

12 Gender-aware pronouns

These commands select pronouns based on the `sex` field (`M` → masculine, `F` → feminine, other/empty → neutral).

- `\heshethey`
- `\himherthem` All take a single `{⟨person-id⟩}` argument. Lowercase and capitalized variants are provided.
- `\hishertheirargs`
- `\HeSheThey`
- `\HimHerThem`
- `\HisHerTheir`

13 Source citations

- `\himselfherselfthemselves`
- `\registersource` `\registersource{⟨text⟩}` produces a footnote citation.
- `\registerendnote` `\registerendnote{⟨text⟩}` produces an endnote if the `endnotes` package is

loaded, otherwise falls back to a footnote.

14 Date formatting

`\registerdate` `\registerdate{YYYY-MM-DD}` formats an ISO-8601 date using the current `datetime2` style. Non-ISO input is passed through with a warning.
`\registerdatestyle` `\registerdatestyle{style}` changes the `datetime2` date style mid-document.

15 Shorthands

`\born`
`\died` `\born{<text>}` prints “b. <text>”; `\died` prints “d.”; `\married` prints “m.”;
`\married` `\baptized` prints “bp.”; `\buried` prints “bur.”. `\place{<text>}` prints <text> in
`\baptized` italics; `\dated{<text>}` prints <text> as-is.
`\buried`
`\place`
`\dated`

16 Register section

`registersection (env.)` The `registersection` environment sets up typographic parameters suitable for register prose: paragraph indentation of 1.5em, no paragraph skip, and compact lists. It also defines a local `\generation{<label>}` command for manual generation headings.

17 Validation

`\checkperson` `\checkperson{<id>}` runs validation checks on a single person: verifies the name is non-empty, all listed parents are defined, all dates are in ISO format, and places are not given without corresponding dates. Problems produce package warnings.
`\checkregister` `\checkregister` runs `\checkperson` on every registered person. Call this near `\end{document}` to catch data integrity issues.

18 Implementation

18.1 Package identification and options

```
1 (*package)
2 %% nehgs-register.sty --- NEHGS-style genealogical register
3 %% Copyright 2026 James P. Howard, II <jh@jameshoward.us>
4 %%
5 %% This work may be distributed and/or modified under the
6 %% conditions of the LaTeX Project Public License, either version 1.3c
7 %% of this license or (at your option) any later version.
8 %% The latest version of this license is in
9 %% https://www.latex-project.org/lppl.txt
10 %% and version 1.3c or later is part of all distributions of LaTeX
11 %% version 2008-05-04 or later.
12 %%
13 %% This work has the LPPL maintenance status ‘maintained’.
14 %% The Current Maintainer of this work is James P. Howard, II.
```



```

15
16 \NeedsTeXFormat{LaTeX2e}[2020/10/01]
17 \ProvidesPackage{nehgs-register}[2026/08/10 v1.1 NEHGS-style genealogical register]

Option booleans and key=value option parser.
18 \newif\ifr@index
19 \r@indextrue
20 \newif\ifr@option@abbrevparents
21 \r@option@abbrevparentsfalse
22 \newif\ifr@option@ngsq
23 \r@option@ngsqfalse
24 \newcommand{\r@datestyle}{}
25 \newcommand{\r@optionkey}{}
26 \newcommand{\r@optionvalue}{}
27 \newcommand{\r@datestylekey}{datestyle}
28 \newcommand{\r@datekey}{date}
29 \newcommand{\r@numberingkey}{numbering}
30 \newcommand{\r@splitoption}{}
31 \def\r@splitoption#1=#2\@nil{%
32   \renewcommand{\r@optionkey}{#1}%
33   \renewcommand{\r@optionvalue}{#2}%
34 }
35 \newcommand{\r@ngsqvalue}{ngsq}
36 \newcommand{\r@handleunknownoption}{%
37   \@expandtwoargs\in@{=}{\CurrentOption}%
38   \ifin@
39     \expandafter\r@splitoption\CurrentOption\@nil
40     \ifx\r@optionkey\r@datestylekey
41       \renewcommand{\r@datestyle}{\r@optionvalue}%
42     \else
43       \ifx\r@optionkey\r@datekey
44         \renewcommand{\r@datestyle}{\r@optionvalue}%
45       \else
46         \ifx\r@optionkey\r@numberingkey
47           \ifx\r@optionvalue\r@ngsqvalue
48             \r@option@ngsqtrue
49           \else
50             \PackageWarning{nehgs-register}{Unknown numbering style '\r@optionvalue'}%
51           \fi
52         \else
53           \PackageWarning{nehgs-register}{Unknown option '\CurrentOption'}%
54         \fi
55       \fi
56     \fi
57   \else
58     \PackageWarning{nehgs-register}{Unknown option '\CurrentOption'}%
59   \fi
60 }
61
62 \DeclareOption{index}{\r@indextrue}
63 \DeclareOption{noindex}{\r@indexfalse}
64 \DeclareOption{fullparents}{\r@option@abbrevparentsfalse}
65 \DeclareOption{abbrevparents}{\r@option@abbrevparentstrue}
66 \DeclareOption{ngsq}{\r@option@ngsqtrue}
67 \DeclareOption*{\r@handleunknownoption}

```

```
68 \ProcessOptions\relax
```

18.2 Required packages

```
69 \RequirePackage{xparse}
70 \RequirePackage{enumitem}
71 \RequirePackage{etoolbox}
72 \RequirePackage{keyval}
73 \RequirePackage{xspace}
74 \RequirePackage{datetime2}
75 \RequirePackage{hyperref}
76
77 \ifr@index
78   \RequirePackage{imakeidx}
79   \makeindex
80 \fi
81
82 \makeatletter
```

18.3 Internal helpers

```
83 \newcommand{\r@trimid}[1]{\zap@space#1 \@empty}
84 \newcommand{\r@warn}[1]{\PackageWarning{nehgs-register}{#1}}
85 \newcommand{\r@indexperson}[1]{\ifr@index\index{#1}\fi}
86
87 \newcommand{\r@unknowngeneration}{0}
88
89 \newcommand{\r@csvgadd}[2]{%
90   \ifx#1\@empty
91     \edef#1{#2}%
92   \else
93     \edef#1{#1,#2}%
94   \fi
95 }
```

18.4 Person registry

`\r@rememberperson` Add a person id to the global person list if not already present.

```
96 \newcommand{\r@personlist}{}
97 \newcommand{\r@rememberperson}[1]{%
98   \edef\r@id{\r@trimid{#1}}%
99   \ifx\r@id\@empty\else
100     \ifcsdef{r@registered@\r@id}{}{%
101       \csdef{r@registered@\r@id}{1}%
102       \r@csvgadd{\r@personlist}{\r@id}%
103     }%
104   \fi
105 }
```

`\r@ensureperson` Conditional that branches on whether a person is defined.

```
106 \newcommand{\r@ensureperson}[1]{%
107   \ifcsdef{r@name@#1}{%
108     \@firstoftwo
109   }{%
110     \r@warn{Undefined person ‘#1’}%
111     \@secondoftwo
112   }
```

```

112 }%
113 }

```

`\r@definepersoninternal` Internal 8-argument person definition used by `\defineperson`.

```

114 \newcommand{\r@definepersoninternal}[8]{%
115   \edef\r@id{\r@trimid{#1}}%
116   \ifx\r@id\empty
117     \r@warn{Empty person id ignored}%
118   \else
119     \r@rememberperson{\r@id}%
120     \ifcsdef{r@name@\r@id}{\r@warn{Redefining person '\r@id'}}{}%
121     \csgdef{r@name@\r@id}{#3}%
122     \csgdef{r@parents@\r@id}{#2}%
123     \csgdef{r@preferred@\r@id}{#4}%
124     \csgdef{r@index@\r@id}{#5}%
125     \csgdef{r@lineageparent@\r@id}{#6}%
126     \csgdef{r@sex@\r@id}{#7}%
127     \csgdef{r@birth@\r@id}{#8}%
128     \csgdef{r@birthplace@\r@id}{}%
129     \csgdef{r@death@\r@id}{}%
130     \csgdef{r@deathplace@\r@id}{}%
131     \csgdef{r@baptism@\r@id}{}%
132     \csgdef{r@baptismplace@\r@id}{}%
133     \csgdef{r@burial@\r@id}{}%
134     \csgdef{r@burialplace@\r@id}{}%
135     \csgdef{r@occupation@\r@id}{}%
136     \csgdef{r@marriagecount@\r@id}{0}%
137     \csundef{r@generation@\r@id}%
138     \csundef{r@generation@working@\r@id}%
139   \fi
140 }

```

`\defineperson` Legacy positional person definition.

```

141 \NewDocumentCommand{\defineperson}{mmmm o}{%
142   \IfNoValueTF{#5}{%
143     \r@definepersoninternal{#1}{#2}{#3}{#4}{#3}{}{}{}%
144   }{%
145     \r@definepersoninternal{#1}{#2}{#3}{#4}{#5}{}{}{}%
146   }%
147 }

```

`\definepersonkv` Key-value person definition.

```

148 \define@key{r@person}{name}{\def\r@kv{name}{#1}}
149 \define@key{r@person}{preferred}{\def\r@kv{preferred}{#1}}
150 \define@key{r@person}{parents}{\def\r@kv{parents}{#1}}
151 \define@key{r@person}{index}{\def\r@kv{index}{#1}}
152 \define@key{r@person}{lineage-parent}{\def\r@kv{lineageparent}{#1}}
153 \define@key{r@person}{sex}{\def\r@kv{sex}{#1}}
154 \define@key{r@person}{birth}{\def\r@kv{birth}{#1}}
155 \define@key{r@person}{birth-place}{\def\r@kv{birthplace}{#1}}
156 \define@key{r@person}{death}{\def\r@kv{death}{#1}}
157 \define@key{r@person}{death-place}{\def\r@kv{deathplace}{#1}}
158 \define@key{r@person}{baptism}{\def\r@kv{baptism}{#1}}
159 \define@key{r@person}{baptism-place}{\def\r@kv{baptismplace}{#1}}

```

```

160 \define@key{r@person}{burial}{\def\r@kv@burial{#1}}
161 \define@key{r@person}{burial-place}{\def\r@kv@burialplace{#1}}
162 \define@key{r@person}{occupation}{\def\r@kv@occupation{#1}}
163 \define@key{r@person}{marriage-date}{\def\r@kv@marriagedate{#1}}
164 \define@key{r@person}{marriage-place}{\def\r@kv@marriageplace{#1}}
165
166 \NewDocumentCommand{\definepersonkv}{mm}{%
167   \def\r@kv@name{}%
168   \def\r@kv@preferred{}%
169   \def\r@kv@parents{}%
170   \def\r@kv@index{}%
171   \def\r@kv@lineageparent{}%
172   \def\r@kv@sex{}%
173   \def\r@kv@birth{}%
174   \def\r@kv@birthplace{}%
175   \def\r@kv@death{}%
176   \def\r@kv@deathplace{}%
177   \def\r@kv@baptism{}%
178   \def\r@kv@baptismplace{}%
179   \def\r@kv@burial{}%
180   \def\r@kv@burialplace{}%
181   \def\r@kv@occupation{}%
182   \def\r@kv@marriagedate{}%
183   \def\r@kv@marriageplace{}%
184   \setkeys{r@person}{#2}%
185   \edef\r@id{\r@trimid{#1}}%
186   \ifx\r@id\@empty
187     \r@warn{Empty person id ignored}%
188   \else
189     \r@rememberperson{\r@id}%
190     \ifx\r@kv@name\@empty
191       \r@warn{Person '\r@id' defined without a name}%
192     \fi
193     \ifx\r@kv@index\@empty
194       \let\r@kv@index\r@kv@name
195     \fi
196     \ifcsdef{r@name@\r@id}{\r@warn{Redefining person '\r@id'}}{}%
197     \expandafter\global\expandafter\let\csname r@name@\r@id\endcsname\r@kv@name
198     \expandafter\global\expandafter\let\csname r@parents@\r@id\endcsname\r@kv@parents
199     \expandafter\global\expandafter\let\csname r@preferred@\r@id\endcsname\r@kv@preferred
200     \expandafter\global\expandafter\let\csname r@index@\r@id\endcsname\r@kv@index
201     \expandafter\global\expandafter\let\csname r@lineageparent@\r@id\endcsname\r@kv@lineageparent
202     \expandafter\global\expandafter\let\csname r@sex@\r@id\endcsname\r@kv@sex
203     \expandafter\global\expandafter\let\csname r@birth@\r@id\endcsname\r@kv@birth
204     \expandafter\global\expandafter\let\csname r@birthplace@\r@id\endcsname\r@kv@birthplace
205     \expandafter\global\expandafter\let\csname r@death@\r@id\endcsname\r@kv@death
206     \expandafter\global\expandafter\let\csname r@deathplace@\r@id\endcsname\r@kv@deathplace
207     \expandafter\global\expandafter\let\csname r@baptism@\r@id\endcsname\r@kv@baptism
208     \expandafter\global\expandafter\let\csname r@baptismplace@\r@id\endcsname\r@kv@baptismplace
209     \expandafter\global\expandafter\let\csname r@burial@\r@id\endcsname\r@kv@burial
210     \expandafter\global\expandafter\let\csname r@burialplace@\r@id\endcsname\r@kv@burialplace
211     \expandafter\global\expandafter\let\csname r@occupation@\r@id\endcsname\r@kv@occupation
212     \csgdef{r@marriagecount@\r@id}{0}%
213     \ifx\r@kv@marriagedate\@empty\else

```

```

214 \csgdef{r@marriagecount@r@id}{1}%
215 \expandafter\global\expandafter\let\csname r@marriagedate@r@id @1\endcsname\r@kv@marri
216 \csgdef{r@marriageplace@r@id @1}{}%
217 \fi
218 \ifx\r@kv@marriageplace@empty\else
219 \ifcsdef{r@marriagedate@r@id @1}{%
220 \expandafter\global\expandafter\let\csname r@marriageplace@r@id @1\endcsname\r@kv@ma
221 }{%
222 \csgdef{r@marriagecount@r@id}{1}%
223 \csgdef{r@marriagedate@r@id @1}{}%
224 \expandafter\global\expandafter\let\csname r@marriageplace@r@id @1\endcsname\r@kv@ma
225 }%
226 \fi
227 \csundef{r@generation@r@id}%
228 \csundef{r@generation@working@r@id}%
229 \fi
230 }

```

18.5 Multiple marriage support

\addmarriage

```

231 \define@key{r@marriage}{date}{\def\r@mkv@date{#1}}
232 \define@key{r@marriage}{place}{\def\r@mkv@place{#1}}
233 \define@key{r@marriage}{spouse}{\def\r@mkv@spouse{#1}}
234
235 \NewDocumentCommand{\addmarriage}{mm}{%
236 \def\r@mkv@date{}%
237 \def\r@mkv@place{}%
238 \def\r@mkv@spouse{}%
239 \setkeys{r@marriage}{#2}%
240 \edef\r@id{\r@trimid{#1}}%
241 \ifx\r@id@empty
242 \r@warn{Cannot add marriage to empty person id}%
243 \else
244 \ifcsdef{r@name@r@id}{%
245 \edef\r@mcnt{\csuse{r@marriagecount@r@id}}%
246 \edef\r@newmcnt{\the\numexpr\r@mcnt+1\relax}%
247 \csgdef{r@marriagecount@r@id}{\r@newmcnt}%
248 \expandafter\global\expandafter\let\csname r@marriagedate@r@id @\r@newmcnt\endcsname\r
249 \expandafter\global\expandafter\let\csname r@marriageplace@r@id @\r@newmcnt\endcsname\r
250 \expandafter\global\expandafter\let\csname r@marriagespouse@r@id @\r@newmcnt\endcsname\r
251 }{%
252 \r@warn{Cannot add marriage to undefined person '\r@id'}%
253 }%
254 \fi
255 }

```

\getmarriagecount

```

\getmarriagedate 256 \newcommand{\getmarriagecount}[1]{%
\getmarriageplace 257 \edef\r@id{\r@trimid{#1}}%
\getmarriagespouse 258 \ifcsdef{r@marriagecount@r@id}{\csuse{r@marriagecount@r@id}}{0}%
259 }
260
261 \newcommand{\getmarriagedate}[2]{%

```

```

262 \edef\r@id{\r@trimid{#1}}%
263 \ifcsdef{r@marriagedate@\r@id @#2}{%
264   \ifcempty{r@marriagedate@\r@id @#2}{%
265     \r@warn{No date stored for marriage #2 of '\r@id'}%
266     \textbf{[marriage date unknown]}%
267   }{%
268     \edef\r@mdate{\csuse{r@marriagedate@\r@id @#2}}%
269     \expandafter\registerdate\expandafter{\r@mdate}%
270   }%
271 }{%
272   \r@warn{No marriage #2 for '\r@id'}%
273   \textbf{[no marriage #2]}%
274 }%
275 }
276
277 \newcommand{\getmarriageplace}[2]{%
278   \edef\r@id{\r@trimid{#1}}%
279   \ifcsdef{r@marriageplace@\r@id @#2}{%
280     \ifcempty{r@marriageplace@\r@id @#2}{%
281       \r@warn{No place stored for marriage #2 of '\r@id'}%
282       \textbf{[marriage place unknown]}%
283     }{%
284       \csuse{r@marriageplace@\r@id @#2}%
285     }%
286   }{%
287     \r@warn{No marriage #2 for '\r@id'}%
288     \textbf{[no marriage #2]}%
289   }%
290 }
291
292 \newcommand{\getmarriagespouse}[2]{%
293   \edef\r@id{\r@trimid{#1}}%
294   \ifcsdef{r@marriagespouse@\r@id @#2}{%
295     \ifcempty{r@marriagespouse@\r@id @#2}{%
296       \edef\r@spid{\csuse{r@marriagespouse@\r@id @#2}}%
297       \getpersonname{\r@spid}%
298     }%
299   }{}%
300 }

```

18.6 Name retrieval

```

\getpersonname
\getpersonnameNI 301 \newcommand{\getpersonname}[1]{%
\getpersonpreferred 302   \edef\r@id{\r@trimid{#1}}%
\getpersonabbrevNI 303   \ifcsdef{r@name@\r@id}{%
\shortname 304     \leavevmode
305     \textsc{\csuse{r@name@\r@id}}%
306     \r@indexperson{\csuse{r@index@\r@id}}%
307   }{%
308     \r@warn{Undefined person '\r@id'}%
309     \textbf{[undefined \r@id]}%
310   }%
311 }

```

```

312
313 \newcommand{\getpersonnameNI}[1]{%
314   \edef\r@id{\r@trimid{#1}}%
315   \ifcsdef{r@name@\r@id}{%
316     \textsc{\csuse{r@name@\r@id}}%
317   }{%
318     \r@warn{Undefined person '\r@id'}%
319     \textbf{[undefined \r@id]}%
320   }%
321 }
322
323 \newcommand{\getpersonpreferred}[1]{%
324   \edef\r@id{\r@trimid{#1}}%
325   \ifcsdef{r@preferred@\r@id}{%
326     \ifcsempy{r@preferred@\r@id}{\textsc{\csuse{r@preferred@\r@id}}}%
327   }{}%
328 }
329
330 \newcommand{\getpersonabbrevNI}[1]{%
331   \edef\r@id{\r@trimid{#1}}%
332   \ifcsdef{r@preferred@\r@id}{%
333     \ifcsempy{r@preferred@\r@id}{%
334       \getpersonnameNI{\r@id}%
335     }{%
336       \textsc{\csuse{r@preferred@\r@id}}%
337     }%
338   }{%
339     \getpersonnameNI{\r@id}%
340   }%
341 }
342
343 \newcommand{\shortname}[1]{\getpersonpreferred{#1}}

```

18.7 Vital record accessors

`\r@getvitaldate` Shared helpers for date and place retrieval.

`\r@getvitalplace`

```

344 \newcommand{\r@getvitaldate}[3]{%
345   \edef\r@id{\r@trimid{#1}}%
346   \ifcsdef{r@#2@\r@id}{%
347     \ifcsempy{r@#2@\r@id}{%
348       \r@warn{No #3 date stored for '\r@id'}%
349       \textbf{[#3 date unknown for \r@id]}%
350     }{%
351       \edef\r@thedata{\csuse{r@#2@\r@id}}%
352       \expandafter\registerdate\expandafter{\r@thedata}%
353     }%
354   }{%
355     \r@warn{Undefined person '\r@id' while retrieving #3 date}%
356     \textbf{[undefined \r@id]}%
357   }%
358 }
359
360 \newcommand{\r@getvitalplace}[3]{%
361   \edef\r@id{\r@trimid{#1}}%

```

```

362 \ifcsdef{r@#2@\r@id}{%
363 \ifcsempy{r@#2@\r@id}{%
364 \r@warn{No #3 place stored for '\r@id'}%
365 \textbf{[#3 place unknown for \r@id]}%
366 }{%
367 \csuse{r@#2@\r@id}%
368 }%
369 }{%
370 \r@warn{Undefined person '\r@id' while retrieving #3 place}%
371 \textbf{[undefined \r@id]}%
372 }%
373 }
374
375 \newcommand{\getpersonbirthdate}[1]{\r@getvitaldate{#1}{birth}{birth}}
376 \newcommand{\getpersonbirthplace}[1]{\r@getvitalplace{#1}{birthplace}{birth}}
377 \newcommand{\getpersondeathdate}[1]{\r@getvitaldate{#1}{death}{death}}
378 \newcommand{\getpersondeathplace}[1]{\r@getvitalplace{#1}{deathplace}{death}}
379 \newcommand{\getpersonbaptismdate}[1]{\r@getvitaldate{#1}{baptism}{baptism}}
380 \newcommand{\getpersonbaptismplace}[1]{\r@getvitalplace{#1}{baptismplace}{baptism}}
381 \newcommand{\getpersonburialdate}[1]{\r@getvitaldate{#1}{burial}{burial}}
382 \newcommand{\getpersonburialplace}[1]{\r@getvitalplace{#1}{burialplace}{burial}}

\r@getpersonoccupation
\r@getpersonsex
383 \newcommand{\getpersonoccupation}[1]{%
384 \edef\r@id{\r@trimid{#1}}%
385 \ifcsdef{r@occupation@\r@id}{%
386 \ifcsempy{r@occupation@\r@id}{%
387 \r@warn{No occupation stored for '\r@id'}%
388 \textbf{[occupation unknown for \r@id]}%
389 }{%
390 \csuse{r@occupation@\r@id}%
391 }%
392 }{%
393 \r@warn{Undefined person '\r@id' while retrieving occupation}%
394 \textbf{[undefined \r@id]}%
395 }%
396 }
397
398 \newcommand{\getpersonsex}[1]{%
399 \edef\r@id{\r@trimid{#1}}%
400 \ifcsdef{r@sex@\r@id}{\csuse{r@sex@\r@id}}{%
401 }

```

18.8 Gender-aware pronouns

\r@pronoun Internal dispatcher: selects masculine, feminine, or neutral form.

```

402 \newcommand{\r@pronoun}[4]{%
403 \edef\r@id{\r@trimid{#1}}%
404 \ifcsdef{r@sex@\r@id}{%
405 \edef\r@thesex{\csuse{r@sex@\r@id}}%
406 \def\r@male{M}\def\r@female{F}%
407 \ifx\r@thesex\r@male #2%
408 \else\ifx\r@thesex\r@female #3%
409 \else #4%

```



```

410 \fi\fi
411 }{#4}%
412 }
413
414 \newcommand{\heshethey}[1]{\r@pronoun{#1}{he}{she}{they}}
415 \newcommand{\himherthem}[1]{\r@pronoun{#1}{him}{her}{them}}
416 \newcommand{\hishertheirargs}[1]{\r@pronoun{#1}{his}{her}{their}}
417 \newcommand{\HeSheThey}[1]{\r@pronoun{#1}{He}{She}{They}}
418 \newcommand{\HimHerThem}[1]{\r@pronoun{#1}{Him}{Her}{Them}}
419 \newcommand{\HisHerTheir}[1]{\r@pronoun{#1}{His}{Her}{Their}}
420 \newcommand{\himselfherselfthemselves}[1]{\r@pronoun{#1}{himself}{herself}{themselves}}

```

18.9 Approximate dates

```

421 \newcommand{\circa}[1]{c.\ #1}
422 \newcommand{\before}[1]{bef.\ #1}
423 \newcommand{\after}[1]{aft.\ #1}
424 \newcommand{\between}[2]{bet.\ #1 and #2}
425 \newcommand{\aboutdate}[1]{abt.\ #1}
426 \newcommand{\probably}[1]{prob.\ #1}
427 \newcommand{\say}[1]{say #1}

```

18.10 Generation computation

`\r@computegeneration` Recursive computation with cycle detection via a “working” flag.

```

428 \newcommand{\r@computegeneration}[1]{%
429 \begingroup
430 \edef\r@target{\r@trimid{#1}}%
431 \ifcsdef{r@generation@\r@target}{%
432 }{%
433 \ifcsdef{r@generation@working@\r@target}{%
434 \r@warn{Cycle detected while computing generation for '\r@target'}%
435 \expandafter\xdef\csname r@generation@\r@target\endcsname{\r@unknowngeneration}%
436 }{%
437 \csdef{r@generation@working@\r@target}{1}%
438 \ifcsdef{r@parents@\r@target}{%
439 \edef\r@parents{\csuse{r@parents@\r@target}}%
440 \ifx\r@parents\empty
441 \expandafter\xdef\csname r@generation@\r@target\endcsname{1}%
442 \else
443 \def\r@max{0}%
444 \def\r@checkparent##1{%
445 \edef\r@pid{\r@trimid{##1}}%
446 \ifx\r@pid\empty
447 \r@warn{Empty parent id listed for '\r@target'}%
448 \else
449 \ifcsdef{r@parents@\r@pid}{%
450 \ifcsdef{r@generation@\r@pid}{\r@computegeneration{\r@pid}}%
451 \edef\r@pg{\csuse{r@generation@\r@pid}}%
452 \ifnum\r@pg>\r@max\relax \xdef\r@max{\r@pg}\fi
453 }{%
454 \r@warn{Undefined parent '\r@pid' listed for '\r@target'}%
455 \expandafter\xdef\csname r@generation@\r@pid\endcsname{\r@unknowngeneration}%
456 }%

```

```

457         \fi
458     }%
459     \expandafter\forcsvlist\expandafter\r@checkparent\expandafter{\r@parents}%
460     \edef\r@this{\the\numexpr\r@max+1\relax}%
461     \expandafter\xdef\csname r@generation@\r@target\endcsname{\r@this}%
462     \fi
463 }{%
464     \r@warn{Undefined person ‘\r@target’; assigning generation \r@unknowngeneration}%
465     \expandafter\xdef\csname r@generation@\r@target\endcsname{\r@unknowngeneration}%
466 }%
467 \csundef{r@generation@working@\r@target}%
468 }%
469 }%
470 \endgroup
471 }

```

\getgeneration

```

472 \newcommand{\getgeneration}[1]{%
473     \begingroup
474     \edef\r@target{\r@trimid{#1}}%
475     \r@computegeneration{\r@target}%
476     \xdef\r@generationresult{\csuse{r@generation@\r@target}}%
477 \endgroup
478 \r@generationresult
479 }

```

18.11 Ancestor chain formatting

\r@collectancestors

```

\r@addoneancestor 480 \newcommand{\r@alist}{}
481 \newif\ifr@ancestorchain@lineage
482
483 \newcommand{\r@collectancestors}[1]{%
484     \edef\r@id{\r@trimid{#1}}%
485     \ifcsdef{r@ancestor@seen@\r@id}{%
486         \r@warn{Cycle detected while collecting ancestors for ‘\r@id’}%
487     }{%
488         \csdef{r@ancestor@seen@\r@id}{1}%
489         \ifcsdef{r@parents@\r@id}{%
490             \ifcsdef{r@lineageparent@\r@id}{%
491                 \ifcempty{r@lineageparent@\r@id}{%
492                     \edef\r@plist{\csuse{r@parents@\r@id}}%
493                     \ifx\r@plist\@empty\else
494                         \expandafter\forcsvlist\expandafter\r@addoneancestor\expandafter{\r@plist}%
495                     \fi
496                 }{%
497                     \r@ancestorchain@lineagetrue
498                     \edef\r@lineagepid{\csuse{r@lineageparent@\r@id}}%
499                     \expandafter\r@addoneancestor\expandafter{\r@lineagepid}%
500                 }%
501             }{%
502                 \edef\r@plist{\csuse{r@parents@\r@id}}%
503                 \ifx\r@plist\@empty\else
504                     \expandafter\forcsvlist\expandafter\r@addoneancestor\expandafter{\r@plist}%

```

```

505     \fi
506   }%
507 }{%
508   \r@warn{Undefined person '\r@id' while collecting ancestors}%
509 }%
510 }%
511 }
512 \newcommand{\r@addoneancestor}[1]{%
513   \edef\r@pid{\r@trimid{#1}}%
514   \ifx\r@pid\@empty
515     \r@warn{Empty parent id in ancestor chain}%
516   \else
517     \ifcsdef\r@parents@\r@pid{%
518       \r@csvadd{\r@alist}{\r@pid}%
519       \r@collectancestors{\r@pid}%
520     }{%
521       \r@warn{Undefined parent '\r@pid' in ancestor chain}%
522       \r@csvadd{\r@alist}{\r@pid}%
523     }%
524   \fi
525 }

```

\formatparents

```

526 \newif\ifr@abbrevparents
527 \ifr@option@abbrevparents
528   \r@abbrevparentstrue
529 \else
530   \r@abbrevparentsfalse
531 \fi
532
533 \newcount\r@pcnt
534 \newcount\r@i
535
536 \newcommand{\r@namegensNI}[1]{%
537   \mbox{\getpersonnameNI{#1}\textsuperscript{\getgeneration{#1}}}%
538 }
539 \newcommand{\r@abbrevnamegensNI}[1]{%
540   \mbox{\getpersonabbrevNI{#1}\textsuperscript{\getgeneration{#1}}}%
541 }
542
543 \newcommand{\formatparents}[1]{%
544   \begingroup
545     \def\r@alist{}%
546     \r@ancestorchain@lineagefalse
547     \r@collectancestors{#1}%
548     \edef\r@plist{\r@alist}%
549     \ifx\r@plist\@empty
550     \else
551       \r@pcnt=0\relax
552       \def\r@countone##1{\advance\r@pcnt by 1\relax}%
553       \expandafter\forcsvlist\expandafter\r@countone\expandafter{\r@plist}%
554       \unskip\space(
555       \r@i=0\relax
556       \def\r@printone##1{%

```

```

557         \edef\r@pid{\r@trimid{##1}}%
558         \advance\r@i by 1\relax
559         \ifnum\r@i>1
560             \ifr@ancestorchain@lineage
561             ,\space
562         \else
563             \ifnum\r@i=\r@pcnt
564                 \space and\space
565             \else
566                 ,\space
567             \fi
568         \fi
569     \fi
570     \ifr@abbrevparents
571         \r@abbrevnamegensNI{\r@pid}%
572     \else
573         \r@namegensNI{\r@pid}%
574     \fi
575 }%
576 \expandafter\forcsvlist\expandafter\r@printone\expandafter{\r@plist}%
577 )%
578 \fi
579 \endgroup
580 }

```

18.12 Main register entries

\mainperson

```

581 \newcommand{\registerpersonsep}{1em}
582 \newif\ifr@afterheading
583 \r@afterheadingfalse
584
585 \pretocmd{\section}{\global\r@afterheadingtrue}{}{}
586 \pretocmd{\subsection}{\global\r@afterheadingtrue}{}{}
587 \pretocmd{\subsubsection}{\global\r@afterheadingtrue}{}{}
588 \pretocmd{\paragraph}{\global\r@afterheadingtrue}{}{}
589 \pretocmd{\subparagraph}{\global\r@afterheadingtrue}{}{}
590
591 \newcounter{registerperson}
592
593 \NewDocumentCommand{\mainperson}{m}{%
594     \edef\r@id{\r@trimid{#1}}%
595     \par
596     \ifr@afterheading
597         \global\r@afterheadingfalse
598     \else
599         \addvspace{\registerpersonsep}%
600     \fi
601     \refstepcounter{registerperson}%
602     \csgdef{\r@mainentry@\r@id}{\theregisterperson}%
603     \label{person:\r@id}%
604     \noindent
605     \textbf{\theregisterperson.\ }%
606     \getpersonname{\r@id}\textsuperscript{\getgeneration{\r@id}}%

```

```

607 \formatparents{\r@id}\xspace
608 }

```

NGSQ numbering override.

```

609 \ifroption@ngsq
610 \renewcommand{\theregisterperson}{\arabic{registerperson}}
611 \fi

```

18.13 Cross-references

```

\xrefperson
\childname 612 \NewDocumentCommand{\xrefperson}{m}{%
613 \edef\r@id{\r@trimid{#1}}%
614 \hyperref[person:\r@id]{%
615 \getpersonname{\r@id}\textsuperscript{\getgeneration{\r@id}}%
616 \space(\#\ref{person:\r@id})%
617 }%
618 }
619
620 \NewDocumentCommand{\childname}{m}{%
621 \edef\r@id{\r@trimid{#1}}%
622 \getpersonnameNI{\r@id}\textsuperscript{\getgeneration{\r@id}}%
623 }

```

18.14 Child lists

```

children (env.)
childrenof (env.) 624 \newif\ifrochildrenof
625 \r@childrenoffalse
626 \newcommand{\r@childrenof@a}{ }
627 \newcommand{\r@childrenof@b}{ }
628
629 \NewDocumentEnvironment{children}{ 0 }{ }{%
630 \begin{enumerate}[label=\roman*. , leftmargin=2em, itemsep=0pt, topsep=0pt, #1]%
631 }\end{enumerate}}
632
633 \NewDocumentEnvironment{childrenof}{ m m 0 }{ }{%
634 \begingroup
635 \edef\r@childrenof@a{\r@trimid{#1}}%
636 \edef\r@childrenof@b{\r@trimid{#2}}%
637 \r@childrenoftrue
638 \begin{children}[#3]%
639 }\end{children}%
640 \endgroup
641
642 }

```

```

\r@checkchildparents
\child 643 \newcommand{\r@checkchildparents}[1]{%
\childref 644 \ifrochildrenof
645 \edef\r@childid{\r@trimid{#1}}%
646 \ifcsdef{r@parents@\r@childid}{%
647 \edef\r@childparents{\csuse{r@parents@\r@childid}}%

```

```

648 \edef\r@expectedparents{\r@childrenof@a,\r@childrenof@b}%
649 \edef\r@expectedparentsrev{\r@childrenof@b,\r@childrenof@a}%
650 \ifx\r@childparents\r@expectedparents
651 \else
652 \ifx\r@childparents\r@expectedparentsrev
653 \else
654 \r@warn{Child '\r@childid' listed under parents '\r@expectedparents' but has parent
655 \fi
656 \fi
657 }{%
658 \r@warn{Undefined child '\r@childid' listed in childrenof}%
659 }%
660 \fi
661 }
662
663 \NewDocumentCommand{\child}{m m}{%
664 \r@checkchildparents{#1}%
665 \item \childname{#1}%
666 \IfBlankF{#2}{, #2}%
667 }
668
669 \NewDocumentCommand{\childref}{m m}{%
670 \r@checkchildparents{#1}%
671 \item \textbf{+}\space\childname{#1}%
672 \IfBlankF{#2}{, #2}%
673 }

```

18.15 Source and note support

```

\registersource
\registerendnote 674 \NewDocumentCommand{\registersource}{m}{%
675 \footnote{#1}%
676 }
677
678 \NewDocumentCommand{\registerendnote}{m}{%
679 \@ifpackageloaded{endnotes}{\endnote{#1}}{\footnote{#1}}%
680 }

```

18.16 Automatic generation headings

```

\generationheading
681 \newcommand{\r@generationword}[1]{%
682 \ifcase#1 Zeroth\or First\or Second\or Third\or Fourth\or Fifth\or
683 Sixth\or Seventh\or Eighth\or Ninth\or Tenth\or Eleventh\or
684 Twelfth\or Thirteenth\or Fourteenth\or Fifteenth\or Sixteenth\or
685 Seventeenth\or Eighteenth\or Nineteenth\or Twentieth\else
686 #1th\fi
687 }
688
689 \newcommand{\generationheading}[1]{%
690 \edef\r@id{\r@trimid{#1}}%
691 \r@computegeneration{\r@id}%
692 \edef\r@gen{\csuse{r@generation@}\r@id}}%

```

```

693 \par\medskip\noindent
694 \textbf{\r@generationword{\r@gen}\space Generation}%
695 \par\nobreak
696 }

```

18.17 Date formatting

```

\registerdate
\registerdatestyle 697 \newcommand{\registerdatestyle}[1]{%
698   \DTMsetdatestyle{#1}%
699 }
700
701 \AtBeginDocument{%
702   \ifdefempty{\r@datestyle}{\DTMsetdatestyle{\r@datestyle}}%
703 }
704
705 \ExplSyntaxOn
706 \cs_new_protected:Npn \nehgs_register_date:n #1
707 {
708   \regex_match:nnTF { \A \d{4} - \d{2} - \d{2} \Z } {#1}
709   { \DTMdate{#1} }
710   {
711     \PackageWarning{nehgs-register}{Date~‘#1’~is~not~in~YYYY-MM-DD~form}
712     #1
713   }
714 }
715
716 \NewDocumentCommand{\registerdate}{m}{\nehgs_register_date:n{#1}}
717 \ExplSyntaxOff

```

18.18 Date validation

```

718 \ExplSyntaxOn
719 \cs_generate_variant:Nn \regex_match:nnF { neF }
720 \cs_new_protected:Npn \nehgs_register_validate_iso_date:nn #1 #2
721 {
722   \tl_if_blank:nF {#2}
723   {
724     \regex_match:neF { \A \d{4} - \d{2} - \d{2} \Z } {#2}
725     { \PackageWarning{nehgs-register}{#1~date~‘#2’~is~not~in~YYYY-MM-DD~form} }
726   }
727 }
728 \NewDocumentCommand{\r@validateisodate}{m m}
729 {\nehgs_register_validate_iso_date:nn {#1} {#2}}
730 \ExplSyntaxOff

```

18.19 Validation commands

```

\checkperson
\checkregister 731 \newcommand{\r@checkdatefield}[3]{%
732   \ifcsdef{\r@#3@#1}{%
733     \edef\r@datevalue{\csuse{\r@#3@#1}}%
734     \ifx\r@datevalue\@empty\else
735       \r@validateisodate{#2 for ‘#1’}{\r@datevalue}%

```

```

736     \fi
737   }{}%
738 }
739
740 \newcommand{\r@checkplacewithoutdate}[5]{%
741   \ifcsdef{r@#2@#1}{%
742     \ifcempty{r@#2@#1}{%
743       \ifcsdef{r@#3@#1}{%
744         \ifcempty{r@#3@#1}{\r@warn{Person '#1' has #4 but no #5}}{}%
745       }\r@warn{Person '#1' has #4 but no #5}}%
746     }%
747   }{}%
748 }
749
750 \newcommand{\r@checkparentlist}[1]{%
751   \ifcsdef{r@parents@#1}{%
752     \edef\r@plist{\csuse{r@parents@#1}}%
753     \ifx\r@plist\@empty\else
754       \def\r@checkoneparent##1{%
755         \edef\r@pid{\r@trimid{##1}}%
756         \ifx\r@pid\@empty
757           \r@warn{Person '#1' has an empty parent id}%
758         \else
759           \ifcsdef{r@name@\r@pid}{\r@warn{Person '#1' lists undefined parent '\r@pid'}}%
760         \fi
761       }%
762       \expandafter\forcsvlist\expandafter\r@checkoneparent\expandafter{\r@plist}%
763     \fi
764   }\r@warn{Person '#1' has no parent field}}%
765 }
766
767 \NewDocumentCommand{\checkperson}{m}{%
768   \begingroup
769     \edef\r@id{\r@trimid{#1}}%
770     \ifx\r@id\@empty
771       \r@warn{Cannot check an empty person id}%
772     \else
773       \ifcsdef{r@name@\r@id}{%
774         \ifcempty{r@name@\r@id}{\r@warn{Person '\r@id' has an empty name}}{}%
775         \r@checkparentlist{\r@id}%
776         \r@checkdatefield{\r@id}{Birth}{birth}%
777         \r@checkdatefield{\r@id}{Death}{death}%
778         \r@checkdatefield{\r@id}{Baptism}{baptism}%
779         \r@checkdatefield{\r@id}{Burial}{burial}%
780         \r@checkplacewithoutdate{\r@id}{birthplace}{birth}{birth place}{birth date}%
781         \r@checkplacewithoutdate{\r@id}{deathplace}{death}{death place}{death date}%
782         \r@checkplacewithoutdate{\r@id}{baptismplace}{baptism}{baptism place}{baptism date}%
783         \r@checkplacewithoutdate{\r@id}{burialplace}{burial}{burial place}{burial date}%
784       }{%
785         \r@warn{Cannot check undefined person '\r@id'}%
786       }%
787     \fi
788   \endgroup
789 }

```



```

790
791 \NewDocumentCommand{\checkregister}{-}{%
792   \begingroup
793     \ifx\r@personlist\@empty
794       \r@warn{No registered persons to check}%
795     \else
796       \def\r@checkoneperson##1{\checkperson{##1}}%
797       \expandafter\forcsvlist\expandafter\r@checkoneperson\expandafter{\r@personlist}%
798     \fi
799   \endgroup
800 }

```

18.20 Shorthands

```

801 \newcommand{\born}[1]{b.\ #1}
802 \newcommand{\died}[1]{d.\ #1}
803 \newcommand{\married}[1]{m.\ #1}
804 \newcommand{\baptized}[1]{bp.\ #1}
805 \newcommand{\buried}[1]{bur.\ #1}
806 \newcommand{\place}[1]{\textit{#1}}
807 \newcommand{\dated}[1]{#1}

```

18.21 Register section environment

`registersection` (*env.*)

```

808 \NewDocumentEnvironment{registersection}{0}{-}{%
809   \begingroup
810   \setlength{\parindent}{1.5em}
811   \setlength{\parskip}{0pt}
812   \setlist{nosep}
813   \def\generation##1{\par\medskip\noindent\textbf{##1 Generation}\par\nobreak}%
814 }{%
815   \par\endgroup
816 }

```

18.22 Backward compatibility

Old public names retained as aliases for one release cycle.

```

817 \let\collectancestors\r@collectancestors
818 \let\addoneancestor\r@addoneancestor
819 \newif\ifabbrevparents
820 \ifr@abbrevparents\abbrevparentstrue\else\abbrevparentsfalse\fi
821
822 \makeatother
823 \endpackage

```

Change History

v1.0

v1.0.1

General: First public release. 1

General: Fix CI and test
infrastructure for TeX Live

2026. 1	v1.1
v1.0.2	General: Include example
General: Rename package to	documents in the CTAN
nehgs-register for CTAN. 1	archive. 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.

Symbols	\buried 805	\CurrentOption 37, 39, 53, 58
\# 616	C	D
\, 421	\checkperson <u>731</u>	\d 708, 724
\@empty 83, 90, 99, 116, 186, 190, 193, 213, 218, 241, 440, 446, 493, 503, 514, 549, 734, 753, 756, 770, 793	\checkregister <u>731</u>	\dated 807
\@expandtwoargs ... 37	\child <u>643</u>	\DeclareOption . 62, 63, 64, 65, 66, 67
\@firstoftwo 108	\childname <u>612</u> , 665, 671	\def 31, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 231, 232, 233, 236, 237, 238, 406, 443, 444, 545, 552, 556, 754, 796, 813
\@ifpackageloaded . 679	\childref <u>643</u>	\define@key ... 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 231, 232, 233
\@nil 31, 39	children (env.) <u>624</u>	\defineperson <u>141</u>
\@secondoftwo 111	childrenof (env.) ... <u>624</u>	\definepersonkv ... <u>148</u>
_ 422, 423, 424, 425, 426, 605, 801, 802, 803, 804, 805	\circa 421	\died 802
A	\collectancestors . 817	\DTMdate 709
\A 708, 724	\cs 706, 719, 720	\DTMsetdatestyle 698, 702
\abbrevparentsfalse 820	\csdef 101, 437, 488	E
\abbrevparentstrue . 820	\csgdef 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 212, 214, 216, 222, 223, 247, 602	\edef 91, 93, 98, 115, 185, 240, 245,
\aboutdate 425	\csname 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 215, 220, 224, 248, 249, 250, 435, 441, 455, 461, 465	
\addmarriage <u>231</u>	\csundef 137, 138, 227, 228, 467	
\addoneancestor ... 818	\csuse 245, 258, 268, 284, 296, 305, 306, 316, 326, 336, 351, 367, 390, 400, 405, 439, 451, 476, 492, 498, 502, 647, 692, 733, 752	
\addvspace 599		
\advance 552, 558		
\after 423		
\arabic 610		
\AtBeginDocument .. 701		
B		
\baptized 804		
\before 422		
\begin 630, 638		
\begingroup 429, 473, 544, 634, 768, 792, 809		
\between 424		
\born 801		

246, 257, 262,	494, 499, 504,	\getpersonnameNI ..
268, 278, 293,	553, 576, 762, 797	 301 , 537, 622
296, 302, 314,	\ExplSyntaxOff 717, 730	\getpersonoccupation
324, 331, 345,	\ExplSyntaxOn . 705, 718	 383
351, 361, 384,		\getpersonpreferred 301
399, 403, 405,	F	\getpersonsex 383
430, 439, 445,	\fi 51, 54,	\global ... 197, 198,
451, 460, 474,	55, 56, 59, 80,	199, 200, 201,
484, 492, 498,	85, 94, 104, 139,	202, 203, 204,
502, 513, 548,	192, 195, 217,	205, 206, 207,
557, 594, 613,	226, 229, 254,	208, 209, 210,
621, 635, 636,	410, 452, 457,	211, 215, 220,
645, 647, 648,	462, 495, 505,	224, 248, 249,
649, 690, 692,	524, 531, 567,	250, 585, 586,
733, 752, 755, 769	568, 569, 574,	587, 588, 589, 597
\else 42, 45, 49, 52, 57,	578, 600, 611,	
92, 99, 118, 188,	655, 656, 660,	H
213, 218, 243,	686, 736, 760,	\HeSheThey 417
408, 409, 442,	763, 787, 798, 820	\heshethey 414
448, 493, 503,	\footnote 675, 679	\HimHerThem 418
516, 529, 550,	\forcsvlist	\himherthem 415
562, 565, 572,	. 459, 494, 504,	\himselfherselfthemselves
598, 651, 653,	553, 576, 762, 797	 420
685, 734, 753,	\formatparents 526 , 607	\HisHerTheir 419
758, 772, 795, 820		\hishertheirargs .. 416
\end 631 , 640	G	\hyperref 614
\endcsname ... 197,	\generation 813	
198, 199, 200,	\generationheading . 681	I
201, 202, 203,	\getgeneration	\ifabbrevparents .. 819
204, 205, 206,	 472 , 537,	\IfBlankF 666, 672
207, 208, 209,	540, 606, 615, 622	\ifcase 682
210, 211, 215,	\getmarriagecount . 256	\ifcsdef 100,
220, 224, 248,	\getmarriagedate .. 256	107, 120, 196,
249, 250, 435,	\getmarriageplace . 256	219, 244, 258,
441, 455, 461, 465	\getmarriagespouse . 256	263, 279, 294,
\endgroup	\getpersonabbrevNI .	303, 315, 325,
. 470, 477, 579,	 301 , 540	332, 346, 362,
641, 788, 799, 815	\getpersonbaptismdate	385, 400, 404,
\endnote 679	 379	431, 433, 438,
environments:	\getpersonbaptismplace	449, 450, 485,
children 624	 380	489, 490, 517,
childrenof 624	\getpersonbirthdate 375	646, 732, 741,
registersection 808	\getpersonbirthplace	743, 751, 759, 773
\expandafter	 376	\ifcsemtpty 264, 280,
.. 39, 197, 198,	\getpersonburialdate	295, 326, 333,
199, 200, 201,	 381	347, 363, 386,
202, 203, 204,	\getpersonburialplace	491, 742, 744, 774
205, 206, 207,	 382	\ifdefemtpy 702
208, 209, 210,	\getpersondeathdate 377	\ifin@ 38
211, 215, 220,	\getpersondeathplace	\IfNoValueTF 142
224, 248, 249,	 378	\ifnum 452, 559, 563
250, 269, 352,	\getpersonname	\ifr@abbrevparents .
435, 441, 455,	. 297, 301 , 606, 615	 526, 570, 820
459, 461, 465,		

<code>\ifr@afterheading</code> .	344, 360, 375,	<code>\r@abbrevparentsfalse</code>	
..... 582, 596	376, 377, 378,	 530	
<code>\ifr@ancestorchain@lineage</code>	379, 380, 381,	<code>\r@abbrevparentstrue</code>	
..... 481, 560	382, 383, 398,	 528	
<code>\ifr@childrenof</code> 624, 644	402, 414, 415,	<code>\r@addoneancestor</code> .	
<code>\ifr@index</code> .. 18, 77, 85	416, 417, 418,	 480, 818	
<code>\ifr@option@abbrevparents</code>	419, 420, 421,	<code>\r@afterheadingfalse</code>	
..... 20, 527	422, 423, 424,	 583, 597	
<code>\ifr@option@ngsq</code> 22, 609	425, 426, 427,	<code>\r@afterheadingtrue</code>	
<code>\ifx</code> 40, 43,	428, 472, 480,	 585,	
46, 47, 90, 99,	483, 512, 536,	586, 587, 588, 589	
116, 186, 190,	539, 543, 581,	<code>\r@alist</code> 480,	
193, 213, 218,	626, 627, 643,	518, 522, 545, 548	
241, 407, 408,	681, 689, 697,	<code>\r@ancestorchain@lineagefalse</code>	
440, 446, 493,	731, 740, 750,	 546	
503, 514, 549,	801, 802, 803,	<code>\r@ancestorchain@lineagetrue</code>	
650, 652, 734,	804, 805, 806, 807	 497	
753, 756, 770, 793	<code>\newcount</code> 533, 534	<code>\r@checkchildparents</code>	
<code>\in@</code> 37	<code>\newcounter</code> 591	 643	
<code>\index</code> 85	<code>\NewDocumentCommand</code>	<code>\r@checkdatefield</code> .	
<code>\item</code> 665, 671	 141,	 731,	
	166, 235, 593,	776, 777, 778, 779	
	612, 620, 663,	<code>\r@checkoneparent</code> .	
	669, 674, 678,	 754, 762	
	716, 728, 767, 791	<code>\r@checkoneperson</code> .	
L	<code>\NewDocumentEnvironment</code>	 796, 797	
<code>\label</code> 603	 629, 633, 808	<code>\r@checkparent</code> 444, 459	
<code>\leavevmode</code> 304	<code>\newif</code> 18, 20, 22, 481,	<code>\r@checkparentlist</code> .	
<code>\let</code> 194, 197,	526, 582, 624, 819	 750, 775	
198, 199, 200,	<code>\nobreak</code> 695, 813	<code>\r@checkplacewithoutdate</code>	
201, 202, 203,	<code>\noindent</code> . 604, 693, 813	 740,	
204, 205, 206,	<code>\numexpr</code> 246, 460	780, 781, 782, 783	
207, 208, 209,		<code>\r@childid</code> ... 645,	
210, 211, 215,		646, 647, 654, 658	
220, 224, 248,		<code>\r@childparents</code> ...	
249, 250, 817, 818		. 647, 650, 652, 654	
	O	<code>\r@childrenof@a</code> ...	
	<code>\or</code> .. 682, 683, 684, 685	. 626, 635, 648, 649	
		<code>\r@childrenof@b</code> ...	
	P	. 627, 636, 648, 649	
<code>\mainperson</code> 581	<code>\PackageWarning</code> 50,	<code>\r@childrenoffalse</code> . 625	
<code>\makeatletter</code> 82	53, 58, 84, 711, 725	<code>\r@childrenoftrue</code> . 637	
<code>\makeatother</code> 822	<code>\par</code> 595,	<code>\r@collectancestors</code>	
<code>\makeindex</code> 79	693, 695, 813, 815	 480, 547, 817	
<code>\married</code> 803	<code>\paragraph</code> 588	<code>\r@computationgeneration</code>	
<code>\mbox</code> 537, 540	<code>\parindent</code> 810	 428, 475, 691	
<code>\medskip</code> 693, 813	<code>\parskip</code> 811	<code>\r@countone</code> ... 552, 553	
	<code>\place</code> 806	<code>\r@csvadd</code> 89, 102, 518, 522	
	<code>\pretocmd</code> 585,	<code>\r@datekey</code> 28, 43	
N	586, 587, 588, 589	<code>\r@datestyle</code> 24, 41, 44, 702	
<code>\NeedsTeXFormat</code> ... 16	<code>\probably</code> 426	<code>\r@datestylekey</code> . 27, 40	
<code>\nehgs</code> 706, 716, 720, 729	<code>\ProcessOptions</code> ... 68		
<code>\newcommand</code> .. 24, 25,	<code>\ProvidesPackage</code> .. 17		
26, 27, 28, 29,			
30, 35, 36, 83,			
84, 85, 87, 89,			
96, 97, 106, 114,			
256, 261, 277,			
292, 301, 313,			
323, 330, 343,			
	R		
	<code>\r@abbrevnamegensNI</code>		
	 539, 571		

\r@datevalue	332, 333, 334,	\r@kv@parents
.... 733, 734, 735	336, 339, 345,	 150, 169, 198
\r@definepersoninternal	346, 347, 348,	\r@kv@preferred ...
.... 114, 143, 145	349, 351, 355,	 149, 168, 199
\r@ensureperson ... 106	356, 361, 362,	\r@kv@sex . 153, 172, 202
\r@expectedparents .	363, 364, 365,	\r@lineagepid . 498, 499
.... 648, 650, 654	367, 370, 371,	\r@male 406, 407
\r@expectedparentsrev	384, 385, 386,	\r@max 443, 452, 460
..... 649, 652	387, 388, 390,	\r@mcnt 245, 246
\r@female 406, 408	393, 394, 399,	\r@mdate 268, 269
\r@gen 692, 694	400, 403, 404,	\r@mkv@date 231, 236, 248
\r@generationresult	405, 484, 485,	\r@mkv@place
..... 476, 478	486, 488, 489,	 232, 237, 249
\r@generationword .	490, 491, 492,	\r@mkv@spouse
..... 681, 694	498, 502, 508,	 233, 238, 250
\r@getvitaldate ... 344	594, 602, 603,	\r@namegensNI . 536, 573
\r@getvitalplace .. 344	606, 607, 613,	\r@newmcnt ... 246,
\r@handleunknownoption	614, 615, 616,	247, 248, 249, 250
..... 36, 67	621, 622, 690,	\r@ngsqvalue 35, 47
\r@i 534,	691, 692, 769,	\r@numberingkey . 29, 46
555, 558, 559, 563	770, 773, 774,	\r@option@abbrevparentsfalse
\r@id 98, 99, 100, 101,	775, 776, 777,	 21, 64
102, 115, 116,	778, 779, 780,	\r@option@abbrevparentstrue
119, 120, 121,	781, 782, 783, 785	 65
122, 123, 124,	\r@indexfalse 63	\r@option@ngsqfalse 23
125, 126, 127,	\r@indexperson . 85, 306	\r@option@ngsqtrue .
128, 129, 130,	\r@indextrue 19, 62	 48, 66
131, 132, 133,	\r@kv@baptism	\r@optionkey
134, 135, 136,	 158, 177, 207	. 25, 32, 40, 43, 46
137, 138, 185,	\r@kv@baptismplace .	\r@optionvalue . 26,
186, 189, 191,	 159, 178, 208	33, 41, 44, 47, 50
196, 197, 198,	\r@kv@birth 154, 173, 203	\r@parents 439, 440, 459
199, 200, 201,	\r@kv@birthplace ..	\r@pcnt 533, 551, 552, 563
202, 203, 204,	 155, 174, 204	\r@personlist
205, 206, 207,	\r@kv@burial	. 96, 102, 793, 797
208, 209, 210,	 160, 179, 209	\r@pg 451, 452
211, 212, 214,	\r@kv@burialplace .	\r@pid 445,
215, 216, 219,	 161, 180, 210	446, 449, 450,
220, 222, 223,	\r@kv@death 156, 175, 205	451, 454, 455,
224, 227, 228,	\r@kv@deathplace ..	513, 514, 517,
240, 241, 244,	 157, 176, 206	518, 519, 521,
245, 247, 248,	\r@kv@index ... 151,	522, 557, 571,
249, 250, 252,	170, 193, 194, 200	573, 755, 756, 759
257, 258, 262,	\r@kv@lineageparent	\r@plist 492, 493, 494,
263, 264, 265,	 152, 171, 201	502, 503, 504,
268, 272, 278,	\r@kv@marriedate .	548, 549, 553,
279, 280, 281,	. 163, 182, 213, 215	576, 752, 753, 762
284, 287, 293,	\r@kv@marriageplace	\r@printone ... 556, 576
294, 295, 296,	 164,	\r@pronoun 402
302, 303, 305,	183, 218, 220, 224	\r@rememberperson .
306, 308, 309,	\r@kv@name ... 148,	 96, 119, 189
314, 315, 316,	167, 190, 194, 197	\r@spid 296, 297
318, 319, 324,	\r@kv@occupation ..	\r@splitoption 30, 31, 39
325, 326, 331,	 162, 181, 211	

<code>\r@target</code> . 430, 431, 433, 434, 435, 437, 438, 439, 441, 447, 454, 461, 464, 465, 467, 474, 475, 476	464, 486, 508, 515, 521, 654, 658, 744, 745, 757, 759, 764, 771, 774, 785, 794	<code>\shortname</code> 301 <code>\space</code> 554, 561, 564, 566, 616, 671, 694 <code>\subparagraph</code> 589 <code>\subsection</code> 586 <code>\subsubsection</code> 587
<code>\r@thedata</code> . . . 351, 352	<code>\ref</code> 616	
<code>\r@thesex</code> . 405, 407, 408	<code>\refstepcounter</code> . . . 601	
<code>\r@this</code> 460, 461	<code>\regex</code> 708, 719, 724	
<code>\r@trimid</code> . . . 83, 98, 115, 185, 240, 257, 262, 278, 293, 302, 314, 324, 331, 345, 361, 384, 399, 403, 430, 445, 474, 484, 513, 557, 594, 613, 621, 635, 636, 645, 690, 755, 769	<code>\registerdate</code> 269, 352, 697 <code>\registerdatestyle</code> . 697 <code>\registerendnote</code> . . 674 <code>\registerpersonsep</code> 581, 599 <code>registersection</code> (env.) 808 <code>\registersource</code> . . . 674 <code>\relax</code> 68, 246, 452, 460, 551, 552, 555, 558	T <code>\textbf</code> 266, 273, 282, 288, 309, 319, 349, 356, 365, 371, 388, 394, 605, 671, 694, 813 <code>\textit</code> 806 <code>\textsc</code> 305, 316, 326, 336 <code>\textsuperscript</code> 537, 540, 606, 615, 622 <code>\the</code> 246, 460 <code>\theregisterperson</code> 602, 605, 610 <code>\tl</code> 722
<code>\r@unknowngeneration</code> 87, 435, 455, 464, 465	<code>\renewcommand</code> 32, 33, 41, 44, 610	U <code>\unskip</code> 554
<code>\r@validateisodate</code> 728, 735	<code>\RequirePackage</code> 69, 70, 71, 72, 73, 74, 75, 78	X <code>\xdef</code> . 435, 441, 452, 455, 461, 465, 476 <code>\xrefperson</code> 612 <code>\xspace</code> 607
<code>\r@warn</code> 84, 110, 117, 120, 187, 191, 196, 242, 252, 265, 272, 281, 287, 308, 318, 348, 355, 364, 370, 387, 393, 434, 447, 454,	S <code>\say</code> 427 <code>\section</code> 585 <code>\setkeys</code> 184, 239 <code>\setlength</code> . . . 810, 811 <code>\setlist</code> 812	Z <code>\Z</code> 708, 724 <code>\zap@space</code> 83