

The `grpslua` package

<+author+>
<+email+>

<+version+> from <+date+>

Contents

1	Introduction	2
2	Usage	2
3	Implementation	2
3.1	Images	3
3.2	first Caps	3
3.3	People and Gramps	4
3.4	Lists	7
3.5	Utils	9
3.6	Ifs	10
3.7	Counters	11
3.8	Keys (grpskeys)	11
3.9	Date	13
3.10	Event	14
3.11	Ancetsor	17
3.12	Descendence	21
3.13	Descendence text per branche	22
3.14	Descendent Tree	24
3.15	Languages	25

1 Introduction

Put text here.

2 Usage

Put text here.

3 Implementation

```
1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{grpslua} [2024/10/05 v0.1 Coupling of Gramps to LuaLaTeX]
```

Lua

```
3 \RequirePackage{luacode}
4 \RequirePackage{luapackageloader}
```

common

```
5 \RequirePackage{multicol}
6 \RequirePackage[top=1in, bottom=1.25in, left=1.0in, right=1.0in]{geometry}
7 \RequirePackage{babel}
8 \RequirePackage{ifthen}
9 \RequirePackage{mfirstuc}
10 \RequirePackage{xkeyval}
11 \RequirePackage{xstring}
12 \RequirePackage{enumitem}
```

Indexing. We do not want package *imakeidx* together with LTXDOC.

```
13 \ifcsname DocInclude\endcsname\relax\else
14 \usepackage{imakeidx}
15 \makeindex[columns=2, title=Alphabetical Index, intoc]
16 \fi
```

Fonts en pdf

```
17 \RequirePackage{fontawesome5}
18 \RequirePackage{hyperref}
19 \hypersetup{
20   colorlinks=true,
21   linkcolor=blue,
22   citecolor=red,
23   filecolor=magenta,
24   urlcolor=cyan,
25   pdftitle={},
26 }
```

\sethypertarget {\langle link\rangle}{\langle txt\rangle} When hypertarget is already set use hyperlink.

```

27 \newcommand{\sethypertarget}[2]{%
28 \ifcsname HT@\#1\endcsname%
29 \hyperlink{\#1}{\#2}%
30 \else%
31 \hypertarget{\#1}{\textcolor{blue!70!black}{\#2}}%
32 \expandafter\def\csname HT@\#1\endcsname{}% Create a definition
33 \fi%
34 }%

```

Packages for images

```

35 \RequirePackage{tikz}
36 \usetikzlibrary{positioning}
37 \RequirePackage{calc}
38 \RequirePackage{graphicx}
39 \RequirePackage{fancyref}
40 \RequirePackage[tikz]{ocgx2}
41 \RequirePackage[all]{genealogytree}

```

3.1 Images

```

42 \tikzset{%
43 tagbox/.style = {red,thick,solid},%
44 tagtext/.style = {black,thick,solid,rounded corners,fill=white},%
45 tagnumber/.style={yellow, font=\bf, fill=blue}%
46 }%

```

\grpsMedium {*medium_handle*} Make tikzpicture with *gramps_id* or *handle*.

```
47 \def\grpsMedium{\directlua{grps.picture(''\luastring{\#1}')}}%
```

\grpsAllMedia Print all images

```
48 \def\grpsAllMedia{\directlua{grps.allmedia( )}}%
```

3.2 first Caps

\firstcap

```
49 \newcommand{\firstcap}[1]{\directlua{word_first_cap(\luastrings{\#1})}}
```

\startcap

```
50 \newcommand{\startcap}[1]{\directlua{line_first_cap(\luastrings{\#1})}}
```

\pointendline

```
51 \newcommand{\pointendline}[1]{\directlua{point_end_line(\luastrings{\#1})}}
```

```
52 \begin{luacode}
53 function word_first_cap ( s )
```

```

54     s = unicode.utf8.gsub( s , "%s*(%a)(%a*)%s*",
55         function(x,y)
56             return ( unicode.utf8.upper(x) .. unicode.utf8.lower(y).. " " )
57         end )
58     tex.sprint ( string.sub(s,1,-2) )
59 end
60 function line_first_cap ( s )
61     s = unicode.utf8.gsub(s,"^%s*(%a)", function(x) return (unicode.utf8.upper(x)) end)
62     tex.sprint ( s )
63 end
64 function point_end_line(s)
65     s = unicode.utf8.gsub(s, "(%a)%s*$", function(x) return(x..". ") end)
66     tex.sprint(s)
67 end
68 \end{luacode}

```

3.3 People and Gramps

```

69 \newcommand\loadgramps{%
70 \directlua{if not grps then grps = require("gramps") end }%

```

\grampsdatabase {*database name*} Full directory of gramps sql-database.

```

71 \newcommand{\grampsdatabase}[1]{%
72     \loadgramps%
73     \directlua{grps.setdatabase('\luaescapestring{\#1}')%
74 }%

```

\fullname {*person_handle or person_id*} \fullname{I0018} = John Hjalmar Smith

```

75 \newcommand\fullname[1]{%
76     \loadgramps%
77 \directlua{tex.print(grps.fullname('\luaescapestring{\#1}'))}%
78 }%

```

\grpsAllPeople Print all people in a list.

```

79 \def\grpsAllPeople{%
80 \noindent
81 \directlua{
local i=1

82 for h,p in pairs(grps.allperson())do
83 % i=i+1
84 %         if i>20 then break end
85 %
86         tex.print("\makebox[0.1\textwidth][1]{"..p.gramps_id .."}")

tex.print(
makebox[0.4
textwidth][l]..h['handle'] ..")
87         tex.print("\makebox[0.28\textwidth][1]{"..p.given_name .."}")

```

```

88     tex.print("\\makebox[0.21\\textwidth][l]{..p.surname ..}")
89     if p.birth and p.birth[3] then
90         tex.print("\\makebox[0.1\\textwidth][l]{..p.birth[3]..}")
91     elseif p.baptism and p.baptism[3] then
92         tex.print("\\makebox[0.1\\textwidth][l]{..p.baptism[3]..}")
93     else
94         tex.print("\\makebox[0.1\\textwidth][l]{--}")
95     end
96     if p.death and p.death[3] then
97         tex.print("\\makebox[0.1\\textwidth][l]{..p.death[3]..}")
98     elseif p.burial and p.burial[3] then
99         tex.print("\\makebox[0.1\\textwidth][l]{..p.burial[3]..}")
100    else
101        tex.print("\\makebox[0.1\\textwidth][l]{--}")
102    end
103    tex.print("\\\\")
104 end
105 }}%

```

\grpsFirstPerson {</person_handle or person_id>} \grpsFirstPerson{I0018} =

John Hjalmar Smith. He 1-30-1932:San Francisco, San Francisco Co., CA. Son of Hjalmar Smith and Marjorie Ohman. He 6-4-1954:Sparks, Washoe Co., NV with Alice Paula Perkins. She 11-22-1933:Sparks, Washoe Co., NV.

- i Edwin Michael. He 5-24-1961:San Jose, Santa Clara Co., CA. He . He 1979- 1984:UC Berkeley, 1984. He 10-5-1994:San Francisco, San Francisco Co., CA, 5-27-1995:San Ramon, Conta Costa Co., CA with Janice Ann Adams. She 8-26-1965:Fremont, Alameda Co., CA. She . She 1988.
 - a Amber Marie. She 4-12-1998:Hayward, Alameda Co., CA.
 - b Mason Michael. He 6-26-1996:Hayward, Alameda Co., CA.
- ii Marjorie Alice. She 2-5-1960:San Jose, Santa Clara Co., CA.

```

106 \newcommand{\grpsFirstPerson}[1]{%
107   \loadgramps
108   \directlua{%
109   if grps.shortformat then
110     grps.short_print_person('\\luaescapestring{#1}',grps.gendepth)
111   else
112     grps.long_print_person('\\luaescapestring{#1}',grps.gendepth)
113   end
114 }%
115 }

```

Output options	in Lua	
grps.OPTION.	type	option
O_CENTRAL=1		LONG=0
O_SPOUSE=2		LIFEVENTS=1
O_PARENTS=4		, VOCATIONAL=2
O_CHILDREN=3		RESIDENCE=3
O_SPOUSE_HANDLE=5		RELATIONS=4
O_LEVEL=6		PARENTS=5
O_MAXMEDIA=7		CHILDREN=6
		FULLNAME=7
		SPOUSEPARENTS=8
		MEDIA=9
		ITEMIZE=10
		SIBLINGS=11

```

116 \begin{luacode}
117 if not grps then grps = require("gramps") end
118 function options_no(opts,no_opts)
119     for i=1,4 do opts[i] = grps.remove_options(opts[i],no_opts) end
120 end
121 function options_yes(opts,no_opts,tot)
122     local stop = tot or 4
123     for i=1,stop do opts[i] = grps.set_options(opts[i],no_opts) end
124 end
125 \end{luacode}

```

\grpsPrintPerson {*gramps_id or handle*} [*option*]
 default
 fullname (1-4)
 depth
 long
 short
 relations
 lifeevents
 parents
 children
 vocation
 residence
 media
 maxmedia
 itemize
 \rpsPrintPerson[fullname,depth=1,long,lifeevents=3,parents=2]{*I0018*}=
 John Hjalmar Smith. He is born at January 30 1932 in San Francisco, San Francisco Co., CA (Birth of John Hjalmar Smith). Son of [Hjalmar Smith](#) and [Marjorie Ohman](#).

1 [Marjorie Lee Smith](#). She is born at November 4 1934 in Reno, Washoe Co., NV (Birth of Marjorie Lee Smith).

2

```

126 \newcommand{\grpsPrintPerson}[2] []{%
127   \loadgramps%
128   \setkeys{grpskeys}{depth=2,fullname,#1}

```

```

129 \directlua{%
130   local options
131   \if@grps@default@ options=grps.default_options() \else options={0,0,0,0,0,1} \fi
132   \if@grps@fullname@
133     options_yes(options,grps.OPTION.FULLNAME)
134     grps.remove_options(options[grps.OPTION.O_CHILDREN], grps.OPTION.FULLNAME)
135   \fi%
136   options_yes(options,grps.OPTION.LONG,\grps@option@long)
137   options_yes(options,grps.OPTION.PARENTS,\grps@option@parents)
138   options_yes(options,grps.OPTION.RELATIONS,\grps@option@relations)
139   options_yes(options,grps.OPTION.LIFEVENTS,\grps@option@lifeevents)
140   options_yes(options,grps.OPTION.VOCATIONAL,\grps@option@vocation)
141   options_yes(options,grps.OPTION.RESIDENCE,\grps@option@residence)
142   options_yes(options,grps.OPTION.MEDIA,\grps@option@media)
143   options_yes(options,grps.OPTION.CHILDREN,\grps@option@children)
144   options_yes(options,grps.OPTION.ITEMIZE,\grps@option@itemize)
145   options[grps.OPTION.O_MAXMEDIA]=\grps@option@maxmedia
146   \if@grps@option@siblings@
147     options[OPTION.O_CENTRAL]=grps.set_options(options[OPTION.O_CENTRAL],OPTION.SIBLINGS)%
148   \else
149     options[OPTION.O_CENTRAL]=grps.remove_options(options[OPTION.O_CENTRAL],OPTION.SIBLINGS)%
150   \fi
151   grps.print_person('\luascapestring{#2}',\grps@option@depth,1,options,1)
152 }%

```

3.4 Lists

```
153 %%
```

Env grampslisti

```

154 \newenvironment{grampslisti}{%
155 \begin{list}{}%
156 {%
157   \renewcommand{\makelabel}[1]{\bf\it\makebox[##1]}
158   \setlength{\itemindent}{5ex}%
159   \setlength{\leftmargin}{0pt}%
160   \setlength{\labelwidth}{7ex}%
161   \addtolength{\topsep}{-0.5\parskip}%
162   \%listparindent \normalparindent
163   \setlength{\parsep}{\parskip}%
164   \setlength{\labelsep}{0pt}%
165   \setlength{\itemsep}{0pt}%
166   \setlength{\leftmargin}{2ex}%
167   \setlength{\rightmargin}{0pt}%
168 }%
169 }\end{list}}

```

Env grampslistii

```

170 \newenvironment{grampslistii}{%
171 \begin{list}{}%
172 {%
173   \renewcommand{\makelabel}[1]{\bf\it\makebox[##1]}
174   \setlength{\itemindent}{5ex}%

```

```

175      \setlength{\leftmargin}{0pt}%
176      \setlength{\labelwidth}{7ex}%
177      \addtolength{\topsep}{-0.5\parskip}%
178      \%listparindent \normalparindent
179      \setlength{\parsep}{\parskip}%
180      \setlength{\labelsep}{0pt}%
181      \setlength{\itemsep}{0pt}%
182      \setlength{\leftmargin}{5ex}%
183      \setlength{\rightmargin}{0pt}%
184      }%
185      \small
186 }\{ \end{list}\}

Env grampslisti

187 \renewenvironment{grampslisti}%
188 {\begin{list}{}{%
189   \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
190 }}%
191 {\end{list}\}

Env grampslistii

192 \renewenvironment{grampslistii}%
193 {\begin{list}{}{%
194   \renewcommand{\makelabel}[1]{\bf\makebox{\makeroman{##1}\relax}}%
195   }\small%
196 {\end{list}\}

Env grampslistiiii

197 \newenvironment{grampslistiiii}%
198 {\begin{list}{}{%
199   \renewcommand{\makelabel}[1]{\makebox{\makealphanum{##1}\relax}}%
200   }\footnotesize%
201 {\end{list}\}

Env grampslistiiiiii

202 \newenvironment{grampslistiiiiii}{\begin{list}{}{%
203   \renewcommand{\makelabel}[1]{\makebox{\makeroman{##1}\relax}}}\footnotesize\end{list}\}

Env grampslistiiiiiii

204 \newenvironment{grampslistiiiiiii}{\begin{list}{}{%
205   \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
206   }\small\end{list}\}

207 \newenvironment{grampslistiiiiiiii}{\begin{list}{}{%
208   \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
209   }\footnotesize\end{list}\}

```

```

Env grampslistiiiiii
210 \newenvironment{grampslistiiiiii}{\begin{list}{}{%
211   \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
212 } \footnotesize}{\end{list}}
Env grampslistiiiiii
213 \newenvironment{grampslistiiiiii}{\begin{list}{}{%
214   \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
215 } \tiny}{\end{list}}
Env compactitemize
216 \newenvironment{compactitemize}{%
217 \setlength{\parskip}{0pt}
218 \begin{itemize}[
219   topsep=0pt,
220   partopsep=0pt,
221   parsep=0pt,
222   itemsep=0pt,
223   leftmargin=3ex,
224   rightmargin=0pt,
225   listparindent=0pt,
226   labelwidth=3ex,
227   labelsep=0pt,
228   itemindent=0ex,
229   align=left,
230   after=\vspace{0pt}\vspace{-\baselineskip}%
231 ]}{\end{itemize}}%
232 }%

```

3.5 Utils

```

233 %%
\modulo {<value>}modulo
234 \def\modulo#1#2{\number\numexpr\moduloop{#1}{#2}\relax}
235 \def\truncdiv#1#2{(#1-(#2-1)/2)/#2}
236 \def\moduloop#1#2{#1-\truncdiv{#1}{#2}*#2)}

```

```

\makeromanumber
237 \newcommand{\makeromanumber}[1]{\setcounter{result}{0}%
238 \addtocounter{result}{#1}%
239 \roman{result}\relax%
240 }

```

```
\makealphanumber
```

```

241 \newcommand{\makealphanumeric}[1]{\setcounter{result}{0}%
242 \addtocounter{result}{#1}%
243 \alph{result}\relax%
244 }

```

3.6 Ifs

```

245 \newif\if@grps@option@shortperson@\@grps@option@shortperson@false%
246 \newif\if@grps@option@tree@\@grps@option@tree@false%
247 \newif\if@grps@option@parttree@\@grps@option@parttree@false%
248 \newif\if@grps@adobe@out@\@grps@adobe@out@false%
249 \newif\if@grps@default@\@grps@default@false%
250 \newif\if@grps@fullname@\@grps@fullname@true%
251 \newif\if@grps@option@siblings@\@grps@option@siblings@false%

```

Column and list handling

```

252 \newif\if@grps@mode@list@\@grps@mode@list@false%
253 \newif\if@columnmode \@columnmodefalse%

```

\startmulticolumn

```

254 \newcommand{\startmulticolumn}{%
255     \if@columnmode\relax\else\begin{multicols}{\grpscolumns}\@columnmodetrue\fi%
256 }%

```

\stopmulticolumn

```

257 \newcommand{\stopmulticolumn}{%
258     \if@columnmode\end{multicols}\@columnmodefalse\fi%
259 }%

```

\startlist

```

260 \newcommand{\startlist}{\if@grps@mode@list@\relax\else\begin{grampslisti}\@grps@mode@list@true\fi}

```

\stoplist

```

261 \newcommand{\stoplist}{\if@grps@mode@list@\end{grampslisti}\@grps@mode@list@false\fi}%
262 \newcounter{grps@itemize@depth}%
263 \setcounter{grps@itemize@depth}{0}%

```

\grpsItemizeActualdepth

```

264 \def\grpsItemizeActualdepth{1}

```

\startitemize

```

265 \newcommand{\startitemize}{%
266     \ifthenelse{\numexpr\value{grps@itemize@depth} < \grpsItemizeActualdepth}{%
267         \stepcounter{grps@itemize@depth}\begin{compactitemize}}{}%
268 }%

```

```

\stopitemize

269 \newcommand{\stopitemize}[1]{%
270     \def\grpsItemizeActualdepth{#1}%
271     \whiledo{\numexpr\value{grps@itemize@depth} = \grpsItemizeActualdepth}{%
272         {\end{compactitemize}}\addtocounter{grps@itemize@depth}{-1}%
273     \edef\grpsItemizeActualdepth{\arabic{grps@itemize@depth}}%
274 }%
275
276 \doitem
277
278 \newcommand{\doitem}[2][\cdots]{%
279     \ifthenelse{\grps@option@itemize > 0}{%
280         \ifthenelse{\grps@option@itemize < \grpsItemizeActualdepth}{#2}{%
281             \startitemize[item[#1]]}{}%
282     }%
283 }

```

3.7 Counters

```

280 \newcounter{Generation}
281 \newcounter{grpsNumber}
282 \newcounter{NewGeneration}
283 \newcounter{result}

```

3.8 Keys (grpskeys)

```
284 \def\stringtrue{true}
```

gen amount of generations or “all”

```

285 \def\grps@gen@all{all}
286 \define@key{grpskeys}{gen}{\def\grps@key@temp{#1}\ifx\grps@gen@all\grps@key@temp%
287     \def\generations{1000}\loadgramps\directlua{grps.generations = 1000}%
288     \else%
289     \def\generations{#1}\loadgramps\directlua{grps.generations = #1}%
290     \fi}%

```

depth generation depth by (grand)children

```

291 \xdef\grps@option@depth{1}
292 \define@key{grpskeys}{depth}{\xdef\grps@option@depth{#1}%
293     \loadgramps\directlua{grps.gendepth = #1}}

```

genpiece cut into pieces of n generations

```
294 \define@key{grpskeys}{genpiece}[4]{\def\generationpiece{#1}}%
```

col column settings

```
295 \define@key{grpskeys}{col}[2]{\def\grpscolumns{#1}}%
```

show siblings

```
296 \define@key{grpskeys}{siblings}[true]{\def\partiallbo{\#1}%
297     \ifx\partiallbo\stringtrue\@grps@option@siblings@true\else\@grps@option@siblings@false\fi}%
```

show tree

```
298 \define@key{grpskeys}{tree}[true]{\def\partiallbo{\#1}%
299     \ifx\partiallbo\stringtrue\@grps@option@tree@true\else\@grps@option@tree@false\fi}%
```

parttree

```
300 \define@key{grpskeys}{parttree}[true]{%
301     \def\partiallbo{\#1}%
302     \ifx\partiallbo\stringtrue\@grps@option@parttree@true\else\@grps@option@parttree@false\fi}%
303 }
```

abobe output

```
304 \define@key{grpskeys}{adobe}[true]{%
305     \def\partiallbo{\#1}%
306     \ifx\partiallbo\stringtrue\@grps@adobe@out@true\else\@grps@adobe@out@false\fi}%
307 }%
308 \define@key{grpskeys}{noadobe}[true]{\@grps@adobe@out@false}%
```

option long sets show long (0=never,1=central person, 4=all)

```
309 \xdef\grps@option@long{4}
310 \define@key{grpskeys}{long}[4]{\xdef\grps@option@long{\#1}\@grps@option@shortperson@false}%
311 \define@key{grpskeys}{short}[true]{\xdef\grps@option@long{0}\@grps@option@shortperson@true}%
```

option media sets show media (0=never,1=central person, 4=all)

```
312 \xdef\grps@option@media{1}
313 \define@key{grpskeys}{media}[1]{\xdef\grps@option@media{\#1}}%
314 \xdef\grps@option@maxmedia{1}
315 \define@key{grpskeys}{maxmedia}[1]{\xdef\grps@option@maxmedia{\#1}}%
```

option vocation sets show vocation (0=never,1=central person, 4=all)

```
316 \xdef\grps@option@vocation{0}
317 \define@key{grpskeys}{vocation}[1]{\xdef\grps@option@vocation{\#1}}%
```

option residence sets show residence (0=never,1=central person, 4=all)

```
318 \xdef\grps@option@residence{0}
319 \define@key{grpskeys}{residence}[1]{\xdef\grps@option@residence{\#1}}%
```

option default start with default options

```
320 \define@key{grpskeys}{default}[true]{\def\partiallbool{\#1}%
321     \ifx\partiallbool\stringtrue\@grps@default@true\else\@grps@default@false\fi}%
```

option fullname sets show fullname

```
322 \define@key{grpskeys}{fullname}[true]{\def\partiallbool{\#1}%
323     \ifx\partiallbool\stringtrue\@grps@fullname@true\else\@grps@fullname@false\fi}%
```

option parents sets show parents (0=never,1=central person, 4=all)

```
324 \xdef\grps@option@parents{0}
325 \define@key{grpskeys}{parents}[1]{\xdef\grps@option@parents{\#1}}%
```

option relations sets show relations (0=never,1=central person, 4=all)

```
326 \xdef\grps@option@relations{0}
327 \define@key{grpskeys}{relations}[1]{\xdef\grps@option@relations{\#1}}%
```

option lifeevents sets show lifeevents (0=never,1=central person, 4=all)

```
328 \xdef\grps@option@lifeevents{3}
329 \define@key{grpskeys}{lifeevents}[1]{\xdef\grps@option@lifeevents{\#1}}%
```

option children sets show children (0=never,1=central person, 4=all)

```
330 \xdef\grps@option@children{1}
331 \define@key{grpskeys}{children}[1]{\xdef\grps@option@children{\#1}}%
```

option itemize itemize information

```
332 \xdef\grps@option@itemize{0}
333 \define@key{grpskeys}{itemize}[1]{\xdef\grps@option@itemize{\#1}}%
```

3.9 Date

```
334 %%
```

```
\getshortdate {\langle yyyy-m-d\rangle}
\getshortdate{2024-11-1}:
default: 11-1-2024
dutch: 1-11-2024
```

```
335 \newcommand{\getshortdate}[1]{%
336     \def\grpsDyear{}\def\grpsDmonth{}\def\grpsDday{}%
337     \StrCut[1]{#1}{-}\grpsDyear\grpsDtemp%
338     \StrCut[1]{\grpsDtemp}{-}\grpsDmonth\grpsDday%
```

```

339  \iflanguage{dutch}{%
340  \ifx\grpsDyear\empty{}\else%
341  \ifx\grpsDmonth\empty{}\else%
342  \ifx\grpsDday\empty{}\else%
343  \grpsDday-\fi\grpsDmonth-\fi\grpsDyear\fi
344  }{%
345  \ifx\grpsDyear\empty{}\else%
346  \ifx\grpsDmonth\empty{}\else\grpsDmonth-%
347  \ifx\grpsDday\empty{}\else%
348  \grpsDday-\fi\fi\grpsDyear\fi%
349  }%
350 }

\getlongdate {<yyyy-m-d>}
\getlongdate{2024-11-1}:
default: November 1 2024
dutch: 1 November 2024

351 \newcommand{\getlongdate}[1]{%
352  \def\grpsDyear{}\def\grpsDmonth{}\def\grpsDday{}%
353  \StrCut[1]{#1}{-}\grpsDyear\grpsDtemp%
354  \StrCut[1]{\grpsDtemp}{-}\grpsDmonth\grpsDday%
355  \iflanguage{dutch}{%
356  \ifx\grpsDyear\empty{}\else%
357  \ifx\grpsDmonth\empty{}\else%
358  \ifx\grpsDday\empty{}\else%
359  \grpsDday,\fi\grpsTmonth{\grpsDmonth} \fi\grpsDyear\fi
360  }{%
361  \ifx\grpsDyear\empty{}\else%
362  \ifx\grpsDmonth\empty{}\else\grpsTmonth{\grpsDmonth} %
363  \ifx\grpsDday\empty{}\else%
364  \grpsDday\ \fi\fi\grpsDyear\fi%
365  }%
366 }

```

3.10 Event

```

367 %%

\grpsIresidence 
368 \def\grpsIresidence{\small\faIcon{home}}%\ticon{home-outline}

\grpsIeducation 
369 \def\grpsIeducation{\small\faIcon{graduation-cap}}

\grpsIoccupation 
370 \def\grpsIoccupation{\faIcon{hammer}>

\grpsEvent {<1 type>} {<2 clader type>} {<3 modifier>} {<4 quality>} {<5 date1>} {<6 place1>}
{<7 date2>} {<8 place2>} {<9 text>}

```

```
\rpsEvent{19}{0}{3}{0}{2024-11-22}{place}{}{}{text}
```

short: ~ 11-22-2024:place

long: is buried about November 22 2024 in place (text)

```
371 \newcommand{\grpsEvent}[9]{%
372 \if@grps@option@shortperson@%
373 \doitem[\getshorteventtype{#1}]{\getshorteventtype{#1}}{%
374 \ifcase#3{}{\or{$<"}\or{$>"}\or{$sim$}\or{}{\or{}{\else{}{}}}\else{}\fi}%
375 \ifthenelse{#3 < 6}{%
376 \ifx&#5&\relax\else\,,\getshortdate{#5}\fi}%
377 \ifthenelse{#3 > 3}{\,-,\, \ifx&#5&\relax\else\getshortdate{#7}\fi}{}%
378 \ifx&#6&\relax\else{:#6}\fi}%
379 {}%text only
380 \else% Long text person
381 \doitem[\makehijzij{\grpssex}]{}%
382 \getlongeventtype{#1} %
383 \ifcase#3{\grpsTop}\or{\grpsTvoor}\or{\grpsTna}\or{\grpsTrond}%
384 \or{\grpsTvan}\or{\grpsTtussen}\or{}{\else{}{}}\fi}%
385 \ifthenelse{#3 < 6}{%
386 \ifx&#5&\leeg\else\getlongdate{#5}\fi}%
387 \ifthenelse{#3 > 3}{\grpsTtot\ifx&#7&\relax\else\getlongdate{#7}\fi}{}%
388 \ifx&#6&\relax\else\ \grpsTte\,,#6\fi}%
389 \typeout{#9}%
390 \ifx&#9&\relax\else\ (#9)\fi}%
391 }%
392 {#9}%text only
393 \fi}%
394 }
```

\getshorteventtype TYPE

```
395 \newcommand{\getshorteventtype}[1]{%
396 \ifcase#1{CUSTOM}%= 0,
397 \or{\gtrsymMarried}= MARRIAGE = 1,
398 \or{\MARR\_SETTL}= 2,
399 \or{\MARR\_LIC}= 3,
400 \or{\MARR\_CONTR}= 4,
401 \or{\MARR\_BANNS}= 5,
402 \or{\gtrsymEngaged}= ENGAGEMENT = 6,
403 \or{\gtrsymDivorced}= DIVORCE = 7,
404 \or{\DIV\_FILING}= 8,
405 \or{\ANNULMENT}= 9,
406 \or{\gtrsymPartnership}= MARR\_ALT = 10,
407 \or{\ADOPT}= 11,
408 \or{\gtrsymBorn}= BIRTH = 12,
409 \or{\gtrsymDied}= DEATH = 13,
410 \or{\ADULT\_CHRISTEN}= 14,
411 \or{\gtrsymBaptized}= BAPTISM = 15,
412 \or{\BAR\_MITZVAH}= 16,
413 \or{\BAS\_MITZVAH}= 17,
414 \or{\BLESS}= 18,
415 \or{\gtrsymBuried}= BURIAL = 19,
416 \or{\CAUSE\_DEATH}= 20,
417 \or{\CENSUS}= 21,
418 \or{\CHRISTEN}= 22,
419 \or{\CONFIRMATION}= 23,
```

```

420 \or{CREMATION}%
421 \or{DEGREE}%
422 \or{\grpsIeducation}%
423 \or{ELECTED}%
424 \or{EMIGRATION}%
425 \or{FIRST\_COMMUN}%
426 \or{IMMIGRATION}%
427 \or{GRADUATION}%
428 \or{MED\_INFO}%
429 \or{MILITARY\_SERV}%
430 \or{NATURALIZATION}%
431 \or{NOB\_TITLE}%
432 \or{NUM\_MARRIAGES}%
433 \or{\grpsIoccupation}%
434 \or{ORDINATION}%
435 \or{PROBATE}%
436 \or{PROPERTY}%
437 \or{RELIGION}%
438 \or{\grpsIresidence}%
439 \or{RETIREMENT}%
440 \or{WILL}%
441 \else{UNKNOWN (#1)}\fi%
442 }

```

\getlongeventtype

```

443 \newcommand{\getlongeventtype}[1]{
444 \ifcase#1[CUSTOM]%
445 \or{\grpsTtrouwde}%
446 \or{MARR\_SETTL}%
447 \or{MARR\_LIC}%
448 \or{MARR\_CONTR}%
449 \or{MARR\_BANNS}%
450 \or{\grpsTverloofde}%
451 \or{\grpsTscheidde}%
452 \or{DIV\_FILING}%
453 \or{ANNULMENT}%
454 \or{MARR\_ALT}%
455 \or{ADOPT}%
456 \or{\grpsTgeboren}%
457 \or{\grpsToverleed}%
458 \or{ADULT\_CHRISTEN}%
459 \or{\grpsTgedoopt}%
460 \or{BAR\_MITZVAH}%
461 \or{BAS\_MITZVAH}%
462 \or{BLESS}%
463 \or{\grpsTbegraven}%
464 \or{CAUSE\_DEATH}%
465 \or{CENSUS}%
466 \or{CHRISTEN}%
467 \or{CONFIRMATION}%
468 \or{CREMATION}%
469 \or{DEGREE}%
470 \or{\grpsTopleiding}%
471 \or{ELECTED}%
472 \or{EMIGRATION}%
473 \or{FIRST\_COMMUN}%

```

```

474 \or{IMMIGRATION}\% = 30,
475 \or{GRADUATION}\% = 31,
476 \or{MED\_INFO}\% = 32,
477 \or{MILITARY\_SERV}\% = 33,
478 \or{NATURALIZATION}\% = 34,
479 \or{NOB\_TITLE}\% = 35,
480 \or{NUM\_MARRIAGES}\% = 36,
481 \or{\grpsTberoepis}\% OCCUPATION\% = 37,
482 \or{ORDINATION}\% = 38,
483 \or{PROBATE}\% = 39,
484 \or{PROPERTY}\% = 40,
485 \or{RELIGION}\% = 41,
486 \or{\grpsTwoonde}\% RESIDENCE\% = 42,
487 \or{RETIREMENT}\% = 43,
488 \or{WILL}\% = 44
489 \else{UNKNOWN}\fi
490 }

```

3.11 Ancetsor

```

491 %%
492 \def\gtrkv@treeprefix{TREE}
493 \def\gtrkv@treesuffix{}
494 \tikzset{
495 /gtr/id/.code={%
496 \xdef\gtr@gkv@idlink{\#1}%
497 \xdef\gtr@gkv@id{\expandonce\gtrkv@idprefix\unexpanded{\#1}\expandonce\gtrkv@idsuffix}},
498 /gtr/tree prefix/.store in=\gtrkv@treeprefix,
499 /gtr/tree suffix/.store in=\gtrkv@treesuffix,
500 /gtr/thistree/.code ={\xdef\gtr@gkv@thistree{\expandonce\gtrkv@treeprefix\unexpanded{\#1}\expandonce\gtrkv@idsuffix}},
501 /gtr/parenttree/.code={\xdef\gtr@gkv@parenttree{\expandonce\gtrkv@treeprefix\unexpanded{\#1}\expandonce\gtrkv@idsuffix}},
502 /gtr/childtree/.code ={\xdef\gtr@gkv@childtree{\expandonce\gtrkv@treeprefix\unexpanded{\#1}\expandonce\gtrkv@idsuffix}},
503 /gtr/lefttree/.code={\xdef\gtr@gkv@lefttree{\expandonce\gtrkv@treeprefix\unexpanded{\#1}\expandonce\gtrkv@idsuffix}},
504 /gtr/righttree/.code ={\xdef\gtr@gkv@righttree{\expandonce\gtrkv@treeprefix\unexpanded{\#1}\expandonce\gtrkv@idsuffix}},
505 /gtr/haschildren/.style = {%
506 name code = {%
507 \hyperlink{\gtr@gkv@idlink}{\gtrPrintName@full (\gtrDBkekule)}%
508 \% \ifdefined\gtr@gkv@lefttree\showocg{\gtr@gkv@lefttree}\$\leftarrow\$ \fi%
509 \% \gtr@gkv@id
510 },
511 },
512 /gtr/hasparents/.style = {%
513 name code = {%
514 \if@grps@adobe@out@%
515 \showocg{\gtr@gkv@parenttree}{\fbox{\$\uparrow\$}}\else%
516 \hyperlink{GRAPH}\expandonce\gtr@gkv@parenttree{\fbox{\$\uparrow\$}}\fi%
517 \\ \vspace{0mm}%
518 \hyperlink{\gtr@gkv@idlink}{\gtrPrintName@full (\gtrDBkekule)}%
519 },
520 },
521 }

```

Lua function to prepare files for genealogytree base is the gramps_id of the tree starts are the bases of the partial tree

```
522 \begin{luacode*}
```

```

523 function grps_prepare_trees(base,starts,piece)
524 for i, h in ipairs(starts) do
525 local filebase = 'data.'..base_id..'.kwartierstaat.'..gen_piece..". "..i
526 local filename = kpse.find_file(filebase)
527 if filename == nil then
528 tex.print("\typeout{create file "..filebase.."}")
529 local file = io.open(filebase, "w")
530 if file then
531 file:write(grps.ancestor_tree_data(h,gen_piece))
532 file:close()
533 else
534 tex.print("\typeout{Error: Could not open the file.}")
535 end
536 end
537 end
538 end
539 \end{luacode*}

```

\AncestorTree {*person_handle or person_id*} {[*maximum generations*}
\AncestorTree{I0018} = [*gen*] total amount of generations (default=3)

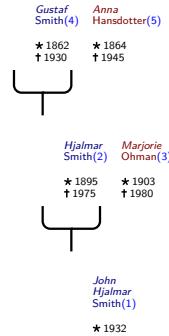


Figure 1: Ancestors of John Hjalmar Smith

[*genpiece=n*] cut the tree in pieces of n generations
[*adobe*] pdf with a lot of scripting best seen in Acrobat Reader
[*noadobe*] default

```

540 \newcommand\AncestorTree[2][]{%
541 \typeout{Ancestortree of: #2}%
542 \setkeys{grpskeys}{gen=3,genpiece=1000,noadobe,#1}%
543 \if@grps@adobe@\directlua{do_adobe=true}\else\directlua{do_adobe=false}\fi%
544 \loadgramps%
545 \directlua{%
546   base_id = "\luaescapestring{\#2}"%
547   local max_gen = \luaescapestring{\generations}%
548   gen_piece = \luaescapestring{\generationpiece}%
549   tree_starts = grps.set_ancestor_tree(base_id,max_gen,gen_piece)%
550   grps_prepare_trees(base_id,tree_starts,gen_piece)%
551   token.set_macro("page@piece@part",grps.format(".3f", (1/(2^(gen_piece-1)))-0.04 ))%
552   \if do_adobe then%
553     tex.print("\begin{figure}[htb]\begin{center}\begin{tikzpicture}"%

```

```

554 tex.print("\node[] (TREEBASE"..base_id..") at (0,0) {};")
555 for i, h in ipairs(tree_starts) do
556 tex.print("\AncestorTreePart"..base_id.."{\generationpiece}{...i...}{...h...}")
557 end
558 tex.print("\end{tikzpicture}\caption{\label{GRAPHTREE#2}\grpsTkwartierstaatvan[#2]}\end{center}")
559 else
560 for i, h in ipairs(tree_starts) do
561 tex.print("\begin{figure}[htb]\begin{center}\hypertarget{GRAPHTREE"..h.."{}{\begin{tikzpicture}")
562 tex.print("\node[] (TREEBASE"..base_id..") at (0,0) {};")
563 tex.print("\AncestorTreePart"..base_id.."{\generationpiece}{...i...}{...h...}")
564 tex.print("\end{tikzpicture}}")
565 tex.print("\caption{\grpsTkwartierstaatvan{\fullname{..h..}}}")
566 tex.print("\end{center}\end{figure}")
567 end
568 end
569 }
570 }

\AncestorTreePart {\langle base id \rangle}
{\langle generation piece \rangle}
{\langle followup number \rangle} start with 1 which is at start visible
{\langle id parent part tree \rangle}

571 \newcommand{\AncestorTreePart}[4]{%
572 \if@grps@adobe@out@%
573 \ifthenelse{#3 = 1}{\def\visibility{on}}{\def\visibility{off}}%
574 \else%
575 \def\visibility{on}%
576 \fi%
577 \typeout{\visibility, #1, #2, #3, #4}%
578 \let\gtr@gkv@childtree\undefined%
579 \let\gtr@gkv@lefttree\undefined%
580 \let\gtr@gkv@righttree\undefined%
581 \begin{scope}[ocg={%
582 name={Tree #3},%
583 ref=TREE#4,%
584 visibility=\visibility,%
585 opts={radiobtngrp=myRadioButtons}%
586 }]\%%
587 \genealogytree[%
588 set position = #4#4 at TREEBASE#1,%
589 template=database traditional,%
590 level size=1.2cm,%
591 level #2-1/.style={level size = 1.7cm},%
592 %level 0/.style={level size = 1.7cm},%
593 node size=\page@piece@part\columnwidth,%2.2
594 id prefix = #4,%
595 date format=yyyy,%
596 list separators={\par}{ }{}{},%
597 name font=\gtrifmale{\selectfont\color{blue!50!black}}{\selectfont\color{red!50!black}},%
598 name code ={\hyperlink{\gtr@gkv@idlink}{\gtrPrintName@full (\gtrDBkekule)}},%
599 ]{\input{data.#1.kwartierstaat.#2.#3}}%
600 \if@grps@adobe@out@%
601 \ifdefinable\gtr@gkv@childtree{\node[draw,blue] (c) at (#4#4.south) {\showocg{\gtr@gkv@childtree}{$}};%
602 \ifdefinable\gtr@gkv@lefttree{\node [draw,blue] (a) at (#4#4.west) {\showocg{\gtr@gkv@lefttree}{$}};%
603 \ifdefinable\gtr@gkv@righttree{\node[draw,blue] (b) at (#4#4.east) {\showocg{\gtr@gkv@righttree}{$}};%

```

```

604 \else%
605 \ifdefined\gtr@gkv@childtree\node[draw,blue] (c) at (#4#4.south) {\hyperlink{GRAPH\expandonce\gtr@%
606 \ifdefined\gtr@gkv@lefttree\node [draw,blue] (a) at (#4#4.west) {\hyperlink{GRAPH\expandonce\gtr@%
607 \ifdefined\gtr@gkv@righttree\node[draw,blue] (b) at (#4#4.east) {\hyperlink{GRAPH\expandonce\gtr@%
608 \fi%
609 %\node[draw,blue] (c) at (#4#4.south west) {Hallo};%
610 \end{scope}%
611 }

\Kwartierstaat  [<options>] {<person_handle or person_id>}%
options: gen=2,depth=1,short/long
\Kwartierstaat[short,gen=4,depth=2]{I0018}

612 \newcommand{\Kwartierstaat}[2] []{%
613 \loadgramps%
614 \typeout{Kwartierstaat van #2}%
615 \setkeys{grpskeys}{gen=2,
616 col=2,
617 genpiece=200,
618 fullname,
619 depth=2,
620 relations=3,
621 long,
622 media=0,
623 tree=false,
624 lifeevents=3,
625 parents=2,
626 #1}
627 \typeout{Start kwartier = \generations, \generationpiece}
628 \directlua{grps.set('luaescapestring{#2}',luaescapestring{\generations})}%
629 \setcounter{Generation}{1}%
630 \setcounter{NewGeneration}{1}%

631 \def\grps@kekule{1}%
632 \ifthenelse{\generations > \generationpiece}{%
633 {\@grps@option@parttree@true}%
634 {\edef\generationpiece{\generations}\@grps@option@parttree@false}%

```

Start the loop

```

635     \@grps@mode@list@false \@columnmodefalse
636 \startmulticolumn
637 \directlua{
638     base_id = "luaescapestring{#2}"
639     new_generation = 2
640     generation = 2
641     local max_gen = luaescapestring{\generations}
642     gen_piece = luaescapestring{\generationpiece}
643     local tree_starts = grps.set_ancestor_tree(base_id,max_gen,gen_piece)
644 local deel_kwartieren = grps.set_kwartier_pieces(tree_starts,gen_piece)
645 \if@grps@option@tree@ grps_prepare_trees(base_id,tree_starts,gen_piece) \fi
646 \if@grps@option@parttree@%
647 token.set_macro("page@piece@part",grps.format(".3f", (1/(2^(gen_piece-1)))-0.04 ),"global")
648 \else
649 token.set_macro("page@piece@part",grps.format(".3f", (1/(2^(max_gen-1)))-0.04 ),"global")
650 \fi

```

```

651 for i, h in ipairs(tree_starts) do
652 %tex.print("\\\typeout{stuk \"..i..\"}")
653 for j,k in pairs(deel_kwartieren[i]) do
654 local handle=grps.kekule_id(k)
655 %tex.print("\\\typeout{stuk \"..i..\" - \"..k..\" ..handle..\"}")
656 if j==1 then
657 \if@grps@option@tree@ %
658 tex.print("\\\stoplist\\\stopmulticolumn")
659 tex.print("\\\begin{figure}[htb]\\begin{center}")
660 tex.print("\\\hypertarget{GRAPHTREE}{\\begin{tikzpicture}}")
661 tex.print("\\\begin{tikzpicture}")
662 tex.print("\\\node[] (TREEBASE\"..base_id..") at (0,0) {};");
663 tex.print("\\\AncestorTreePart\"..base_id..\"{\\\generationpiece}{\"..i..\"}{\"..handle..\"}")
664 tex.print("\\\end{tikzpicture}")
665 tex.print("\\\end{center}\\end{figure}")
666 \fi
667 tex.print("\\\startmulticolumn")
668 tex.print("\\\startlist")
669 tex.print("\\\item[\"..k..\"]")
670 tex.print("\\\hyperlink{\"..handle..\"}{\\begin{tikzpicture}}")
671 \if@grps@option@parttree@%
672 tex.print("\\\addcontentsline{toc}{subsection}{\\begin{tikzpicture}}")
673 \fi%
674 else
675 \if@grps@option@parttree@%
676 \else%
677 if k >= new_generation then
678 tex.print("\\\stoplist")
679 tex.print("\\\subsection*{\\begin{tikzpicture}}")
680 tex.print("\\\addcontentsline{toc}{subsection}{\\begin{tikzpicture}}")
681 new_generation = new_generation*2
682 generation = generation+1
683 end
684 \fi%
685 tex.print("\\\startlist")
686 tex.print("\\\item[\"..k..\"]")
687 print("Preparing person "..k)
688 tex.print("\\\grpsPrintPerson[]{\"..handle..\"}()")
689 end
690 end
691 end
692 tex.print("\\\stoplist\\\stopmulticolumn")
693 }
694 }

```

3.12 Descendence

695 %%

\Descendence [*options*] {*person_handle or person_id*}
options: gen=2,depth=1,short/long, col=2
\Descendence[short,gen=4,depth=2,col=2]{I0018}

```

696 \newcommand{\Descendence}[2] []{%
697     \loadgramps%
698     \typeout{Nageslacht van #2}%

```

```

699  \setkeys{grpskeys}{gen=2,depth=1,col=2,short,#1}
700  \directlua{%
701      grps.set_descendence('\luaescapestring{#2}',\luaescapestring{\generations},true)%
702  \setcounter{Generation}{1}%
703  \setcounter{NewGeneration}{1}%
704  \setcounter{grpsNumber}{1}%
705  \grps@mode@list@false%
706  \columnmodefalse%

```

Start the loop

```

707  \startmulticolumn{%
708  \whiledo{ \numexpr\value{Generation} - 1 < \generations }{%
709      \typeout{printing \arabic{grpsNumber}}%
710      \directlua{%
711          local ghi = grps.get_descendent(\arabic{grpsNumber})
712          tex.print("\setcounter{Generation}{..ghi[1]..}")%
713          tex.print("\gdef\handle{..ghi[2]..}")%
714          tex.print("\gdef\itemindex{..ghi[3]..}")%
715          tex.print("\gdef\grampsid{..ghi[4]..}")%
716      }%
717      \ifthenelse{\numexpr\value{Generation} - 1 < \generations}{%
718          \ifthenelse{\value{NewGeneration} = \value{Generation}}{%
719              \subsection*{Generatie \arabic{Generation}}%
720              \addcontentsline{toc}{subsection}{\arabic{Generation} \arabic{Generation}}%
721              \stepcounter{NewGeneration}{}%
722          \startlist
723          \item[\itemindex] \grpsFirstPerson{\handle}%
724          \stoplist
725      }{}%
726      \stepcounter{grpsNumber}%
727  }%
728  \stoplist%
729  \stopmulticolumn%
730 }

```

3.13 Descendence text per branche

```
731 %%
```

```
\Descendenceb [⟨options⟩] {⟨person_handle or person_id⟩}
options: gen=2,depth=1,short/long, col=2
\Descendenceb[short,gen=4,depth=2,col=2,genpiece=3]{I0018}
```

```

732 \newcommand{\Descendenceb}[2] [] {%
733     \loadgramps%
734     \typeout{Nageslacht van #2}%
735     \setkeys{grpskeys}{gen=2,depth=1,col=2,genpiece=3,short,#1}
736     \directlua{grps.set_descendence('\luaescapestring{#2}',\luaescapestring{\generations},false)%
737     \setcounter{Generation}{1}%
738     \setcounter{NewGeneration}{1}%
739     \setcounter{grpsNumber}{1}%
740     \grps@mode@list@false%
741     \columnmodefalse%
742     \startmulticolumn

```

Start the loop

```
743 \whiledo{ \numexpr\value{Generation} - 1 < \generations }{%
744   \directlua{
745     local ghi = grps.get_descendent(\arabic{grpsNumber})
746     tex.print("\setcounter{Generation}{..ghi[1]..}")
747     tex.print("\gdef\handle{..ghi[2]..}")
748     tex.print("\gdef\itemindex{..ghi[3]..}")
749     tex.print("\gdef\grampsid{..ghi[4]..}")
750   }%
751   \ifthenelse{\numexpr\value{Generation} -1 < \generations}{%
752     \edef\modGeneration{\modulo{\arabic{Generation}}{\generationpiece}}% Generations Modulo
753     \typeout{modgeneration = \modGeneration, \arabic{Generation}}%
754     \ifthenelse{\modGeneration = 1}{%
755       \stoplist%
756       \stopmulticolumn%
757       \begin{center}
758         \pgfmathtruncatemacro{\result}{\generationpiece +1}%
759         \DescendantTree[parttree=true]{\grampsid}{\result}%
760       \end{center}
761     }{\typeout{modGeneration != 1}}%
762     \startmulticolumn
763     \startlist
764     \item[\itemindex] \grpsFirstPerson{\handle}%
765   }{%
766     \stepcounter{grpsNumber}%
767   }%
768   \stoplist%
769   \stopmulticolumn%
770 }
```

Descendant Tree

```
771 %%
```

```
\Descendenceall [<options>] {<person_handle or person_id>}
options: gen=2,depth=1,short/long, col=2
\Descendenceall[short,gen=4,depth=2,col=2,genpiece=3]{I0018}

772 \newcommand{\Descendenceall}[2][]{%
773   \loadgramps%
774   \typeout{Nageslacht van #2}%
775   \setkeys{grpskeys}{gen=2,depth=1,short,genpiece=2,col=2,#1}
776   \directlua{%
777     grps.set_descendence('luaescapestring{#2}',luaescapestring{\generations})}%
778   \setcounter{Generation}{1}%
779   \setcounter{NewGeneration}{1}%
780   \setcounter{grpsNumber}{1}%
781   \startmulticolumn
```

Start the loop

```
782 \whiledo{ \numexpr\value{Generation} - 1 < \generations }{%
783   \directlua{
784     local ghi = grps.get_descendent(\arabic{grpsNumber})
```

```

785     tex.print("\\setcounter{Generation}{..ghi[1]..}")
786     tex.print("\\gdef\\handle{..ghi[2]..}")
787     tex.print("\\gdef\\itemindex{..ghi[3]..}")
788     tex.print("\\gdef\\grampsid{..ghi[4]..}")
789   }%
790   \typeout{gerneration=\arabic{Generation}, \itemindex - \grampsid (\handle)}
791   \ifthenelse{\numexpr\value{Generation} - 1 < \generations}{%
792     \edef\modGeneration{\modulo{\arabic{Generation}}{\generationpiece}}% Generations Modu
793     \typeout{modgeneration = \modGeneration, \arabic{Generation}}%
794     \ifthenelse{\value{NewGeneration} = \value{Generation}}{%
795       \subsection*{Generatie \arabic{Generation}}
796       \addcontentsline{toc}{subsection}{\arabic{Generation} \arabic{Generation}}
797       \stepcounter{NewGeneration}}{%
798       \ifthenelse{\modGeneration = 1}{%
799         \stoplist%
800         \stopmulticolumn%
801         \begin{center}
802           \pgfmathtruncatemacro{\result}{\generationpiece +1}%
803           \DescendentTree[parttree=true]{\grampsid}{\result}%
804         \end{center}
805       }{%
806     }%
807     \startmulticolumn%
808     \startlist%
809     \item[\itemindex] \%arabic{grpsNumber}. \itemindex-\grampsid %, generation=\arabic{Genera
810     \grpsFirstPerson{\handle}%
811     \stepcounter{grpsNumber}}%
812   }%
813   \stoplist%
814   \stopmulticolumn%
815 }%

```

\DescendentTree 3.14 Descendent Tree

```

816 \newcommand\DescendentTree[3][]{%
817   \loadgramps%
818   \typeout{#1,#2,#3}
819   \setkeys{grpskeys}{parttree=true,#1}
820   \directlua{%
821     local id = "\luaescapestring{#2}"
822     local max_gen = \luaescapestring{#3}
823     local command = "\\DescendentTree{..id..}{..max_gen..}"
824     filebase = 'data.'..id..'.descendent.'..max_gen
825     %%    io.write("\string\n In "..command.." gebruikt "..filebase)
826     local filename = kpse.find_file(filebase)
827     %%    if filename == nil then
828       io.write("\string\n In "..command.." creating "..filebase)
829       local file = io.open(filebase, "w")
830       if file then
831         file:write(grps.descendent_tree_data(id,max_gen))
832         file:close()
833         filename=filebase
834       else
835         io.write("\string\n In "..command.." ERROR "..filename.." not OPENED")
836         tex.print("Error: Could not open the file.")

```

```

837         end

end

838     io.write("\string\n In "..command.." using "..filename)
839 }

840 \if@grps@option@parttree@
841 \typeout{parttree true}
842 \tikzset{%
843     /gtr/haschildren/.style={name code ={\hypertarget{CHILD\gtr@gkv@id}{} }\hyperlink{STAM\gtr@gkv@id}{},
844     /gtr/hasparents/.style={name code ={\hypertarget{STAM\gtr@gkv@id}{} }\hyperlink{CHILD\gtr@gkv@id}{},
845     }%
846 \else
847 \typeout{parttree false}
848 \tikzset{%
849     /gtr/haschildren/.style={name code ={\gtrPrintName@full}},%
850     /gtr/hasparents/.style={name code ={\gtrPrintName@full}},%
851     }%
852 \fi
853 \begin{tikzpicture}
854 \genealogytree[%
855     template=database traditional,
856     level size=0.18\columnwidth,
857     level distance = 7mm,
858     node size=6mm,%2.2
859     child distance=-1.5mm,
860     parent distance=-2.5mm,
861     date format=yyyy,
862     timeflow=right,
863     list separators={\newline}{ }{}{},
864     name font=\gtrifmale{\selectfont\color{blue!50!black}}{\selectfont\color{red!50!black}},
865     name code ={\hyperlink{\gtr@gkv@id}{\gtrPrintName@full}},
866     childless/.style={%
867         family={%
868             edges={foreground=red!20!black,line width=0.1mm},xshift=-1mm},
869         },
870     },
871 ]{input{data.#2.descendent.#3}}
872 \end{tikzpicture}
873 }

```

3.15 Languages

English

```

874 \def\grpsThij{he }%
875 \def\grpsTHij{He }%
876 \def\grpsTzij{she }%
877 \def\grpsTZij{She }%
878 \def\grpsTis{is }%
879 \def\grpsTmet{with }%
880 \def\grpsTen{and }%
881 \def\grpsTZoon{Son of }%
882 \def\grpsTDochter{Daughter of }%

```

used in events

```
883 \def\grpsTop{at }
884 \def\grpsTvoor{before }
885 \def\grpsTna{after }
886 \def\grpsTrond{about }
887 \def\grpsTvan{from }
888 \def\grpsTtussen{between }
889 \def\grpsTtrouwde{married }
890 \def\grpsTscheidde{divorced }
891 \def\grpsTverloofde{engaged }
892 \def\grpsTkoppelde{had a relation }
893 \def\grpsTgeboren{is born }
894 \def\grpsToverleed{died }
895 \def\grpsTbegraven{is buried}
896 \def\grpsTgedoopt{is baptizeed }
897 \def\grpsTwoonde{lived }
898 \def\grpsTte{in }
899 \def\grpsTtot{til}
900 \def\grpsTberoepis{occupation is }
901 \def\grpsTopleiding{is educated }
902 \newcommand{\grpsTmonth}[1]{%
903   \ifcase#1%
904     \or January\or February\or March\or April\or May\or June%
905     \or July\or August\or September\or October\or November\or December%
906   \else
907     Invalid month number
908   \fi}%
909 \def\grpsTkwartierstaatvan#1{Ancestors of #1}%
910 \def\grpsTdescendenceof#1{Descendence of #1}%
911 \def\grpsTvoorouder#1{Generation #1}%
912 \newcommand{\makehijzij}[1]{\ifcase#1{zij } \or{hij } \else oo\fi}%
```

Dutch

```
913 \iflanguage{dutch}{%
914   \def\grpsThij{hij }%
915   \def\grpsTHij{Hij }%
916   \def\grpsTzij{zij }%
917   \def\grpsTZij{Zij }%
918   \def\grpsTmet{met }%
919   \def\grpsTen{en }%
920   \def\grpsTZoon{Zoon van }%
921   \def\grpsTDochter{Dochter van }%
```

used in events

```
922 \def\grpsTop{op }
923 \def\grpsTvoor{voor }
924 \def\grpsTna{na }
925 \def\grpsTrond{rond }
926 \def\grpsTvan{van }
927 \def\grpsTtot{tot }
928 \def\grpsTtussen{tussen }
929 \def\grpsTtrouwde{trouwde }
930 \def\grpsTscheidde{scheidde }
```

```

931 \def\grpsTkoppelde{had een relatie }
932 \def\grpsTverloofde{verloofde }
933 \def\grpsTgeboren{is geboren }
934 \def\grpsToverleed{is overleed }
935 \def\grpsTbegraven{is begraven }
936 \def\grpsTgedoopt{is gedoopt }
937 \def\grpsTwoonde{woonde }
938 \def\grpsTte{in }
939 \def\grpsTberoepis{beroep is }
940 \def\grpsTopleiding{is opgeleid }
941 \renewcommand{\grpsTmonth}[1]{%
942 \ifcase#1%
943     \or januarie\or februarie\or maart\or april\or mei\or juni%
944     \or juli\or augustus\or september\or oktober\or november\or december%
945 \else
946     incorrect maand numbrmer
947 \fi}%
948 \def\grpsTkwartierstaatvan#1{Kwartierstaat van #1}
949 \def\grpsTdescendenceof#1{Afstammelingen van #1}%
950 \renewcommand\grpsTvoorouder[1]{%
951 \ifcase#1%
952     \or Kwartierdrager%
953     \or Ouders \or Grootouders \or Overgrootouders \or Betovergrootouders%
954     \or Oudouders \or Oudgrootouders\or Oudovergrootouders \or Oudbetovergrootouders%
955     \or Stamouders \or Stamgrootouders \or Stamovergrootouders \or Stambetovergrootouders%
956     \or Stamoudouders \or Stamoudgrootouders\or Stamoudovergrootouders \or Stamoudbetovergrootouders%
957     \or Edelouders \or Edelgrootouders \or Edelovergrootouders \or Edelbetovergrootouders%
958     \or Edeloudouders \or Edeloudgrootouders\or Edeloudovergrootouders \or Edeloudbetovergrootouders%
959     \or Edelstamouders \or Edelstamgrootouders \or Edelstamovergrootouders \or Edelstambetovergrootouders%
960     \or Edelstamoudouders \or Edelstamoudgrootouders\or Edelstamoudovergrootouders \or Edelstamoudbetovergrootouders%
961     \or Voorouders \or Voorgrootouders \or Voorovergrootouders \or Voorbetovergrootouders%
962     \or Vooroudouders \or Vooroudgrootouders\or Vooroudovergrootouders \or Vooroudbetovergrootouders%
963     \or Voorstamouders \or Voorstamgrootouders \or Voorstamovergrootouders \or Voorstambetovergrootouders%
964     \or Voorstamoudouders \or Voorstamoudgrootouders\or Voorstamoudovergrootouders \or Voorstamoudbetovergrootouders%
965     \or Vooredelouders \or Vooredelgrootouders \or Vooredelovergrootouders \or Vooredelbetovergrootouders%
966     \or Vooredeloudouders \or Vooredeloudgrootouders\or Vooredeloudovergrootouders \or Vooredeloudbetovergrootouders%
967     \or Vooredelstamouders \or Vooredelstamgrootouders \or Vooredelstamovergrootouders \or Vooredelstambetovergrootouders%
968     \or Vooredelstamoudouders \or Vooredelstamoudgrootouders\or Vooredelstamoudovergrootouders \or Vooredelstamoudbetovergrootouders%
969 \else%
970     Generatie #1%
971 \fi}%
972 }{}}%

```