The grpslua package

<+author+>
<+email+>

<+version+> from <+date+>

Contents

1 Introduction

Put text here.

2 Usage

Put text here.

3 Implementation

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

Lua

```
\RequirePackage{luacode}
\RequirePackage{luapackageloader}
```

common

```
\RequirePackage{multicol}
\RequirePackage[top=1in, bottom=1.25in, left=1.0in, right=1.0in]{geometry}
\RequirePackage{babel}
\RequirePackage{ifthen}
\RequirePackage{mfirstuc}
\RequirePackage{xkeyval}
\RequirePackage{xstring}
\RequirePackage{enumitem}
```

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

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

Fonts en pdf

```
\RequirePackage{fontawesome5}
\RequirePackage{hyperref}
\hypersetup{
    colorlinks=true,
    linkcolor=blue,
    citecolor=red,
    filecolor=magenta,
    urlcolor=cyan,
    pdftitle={},
    }
```

`\sethypertarget` `{`⟨*link*⟩`}{`⟨*txt*⟩`}` When hypertarget is already set use hyperlink.

```
\newcommand\sethypertarget[2]{%
\ifcsname HT@#1\endcsname%
\hyperlink{#1}{#2}%
\else%
\hypertarget{#1}{\textcolor{blue!70!black}{#2}}%
\expandafter\def\csname HT@#1\endcsname{}% Create a definition
\fi%
}%
```

Packages for images

```
\RequirePackage{tikz}
\usetikzlibrary{positioning}
\RequirePackage{calc}
\RequirePackage{graphicx}
\RequirePackage{fancyref}
\RequirePackage[tikz]{ocgx2}
\RequirePackage[all]{genealogytree}
```

3.1 Images

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

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

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

`\grpsAllMedia` Print all images

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

3.2 first Caps

`\firstcap`

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

`\startcap`

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

`\pointendline`

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

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

```
   s = unicode.utf8.gsub( s , "%s*(%a)(%a*)%s*",
          function(x,y)
            return ( unicode.utf8.upper(x) .. unicode.utf8.lower(y).." " )
          end )
   tex.sprint ( string.sub(s,1,-2) )
end
function line_first_cap ( s )
   s = unicode.utf8.gsub(s,"^%s*(%a)", function(x) return (unicode.utf8.upper(x)) end)
   tex.sprint ( s )
end
function point_end_line(s)
   s = unicode.utf8.gsub(s, "(%a)%s*$", function(x) return(x..". ") end)
   tex.sprint(s)
end
\end{luacode}
```

3.3 People and Gramps

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

`\grampsdatabase` `{`⟨*database name*⟩`}` Ful directory of gramps sql-database.

```
\newcommand{\grampsdatabase}[1]{%
    \loadgramps%
    \directlua{grps.setdatabase('\luaescapestring{#1}')}
}%
```

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

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

`\grpsAllPeople` Print all people in a list.

```
\def\grpsAllPeople{%
\noindent
\directlua{
```

local i=1

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

tex.print(”
makebox[0.4
textwidth][l]”..h[’handle’] ..””)

```
        tex.print("\\makebox[0.28\\textwidth][l]{"..p.given_name .."}")
```

```
        tex.print("\\makebox[0.21\\textwidth][l]{"..p.surname .."}")
        if p.birth and p.birth[3] then
            tex.print("\\makebox[0.1\\textwidth][l]{"..p.birth[3].."}")
        elseif p.baptism and p.baptism[3] then
            tex.print("\\makebox[0.1\\textwidth][l]{"..p.baptism[3].."}")
        else
            tex.print("\\makebox[0.1\\textwidth][l]{--}")
        end
        if p.death and p.death[3] then
            tex.print("\\makebox[0.1\\textwidth][l]{"..p.death[3].."}")
        elseif p.burial and p.burial[3] then
            tex.print("\\makebox[0.1\\textwidth][l]{"..p.burial[3].."}")
        else
            tex.print("\\makebox[0.1\\textwidth][l]{--}")
        end
      tex.print("\\\\")
    end
}}%
```

`\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.

```
\newcommand{\grpsFirstPerson}[1]{%
  \loadgramps
  \directlua{%
if grps.shortformat then
grps.short_print_person('\luaescapestring{#1}',grps.gendepth)
else
grps.long_print_person('\luaescapestring{#1}',grps.gendepth)
end
}%
}
```

Output options in Lua

grps.OPTION.	type	option
	O_CENTRAL=1	LONG=0
	O_SPOUSE=2	LIFEEVENTS=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

```
\begin{luacode}
  if not grps then grps = require("gramps") end
  function options_no(opts,no_opts)
      for i=1,4 do opts[i] = grps.remove_options(opts[i],no_opts) end
  end
  function options_yes(opts,no_opts,tot)
      local stop = tot or 4
      for i=1,stop do opts[i] = grps.set_options(opts[i],no_opts) end
  end
\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

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

```
\directlua{%
  local options
  \if@grps@default@ options=grps.default_options() \else options={0,0,0,0,0,1} \fi
  \if@grps@fullname@
    options_yes(options,grps.OPTION.FULLNAME)
    grps.remove_options(options[grps.OPTION.O_CHILDREN], grps.OPTION.FULLNAME)
  \fi%
  options_yes(options,grps.OPTION.LONG,\grps@option@long)
  options_yes(options,grps.OPTION.PARENTS,\grps@option@parents)
  options_yes(options,grps.OPTION.RELATIONS,\grps@option@relations)
  options_yes(options,grps.OPTION.LIFEEVENTS,\grps@option@lifeevents)
  options_yes(options,grps.OPTION.VOCATIONAL,\grps@option@vocation)
  options_yes(options,grps.OPTION.RESIDENCE,\grps@option@residence)
  options_yes(options,grps.OPTION.MEDIA,\grps@option@media)
  options_yes(options,grps.OPTION.CHILDREN,\grps@option@children)
  options_yes(options,grps.OPTION.ITEMIZE,\grps@option@itemize)
  options[grps.OPTION.O_MAXMEDIA]=\grps@option@maxmedia
  \if@grps@option@siblings@
    options[OPTION.O_CENTRAL]=grps.set_options(options[OPTION.O_CENTRAL],OPTION.SIBLINGS)%
  \else
    options[OPTION.O_CENTRAL]=grps.remove_options(options[OPTION.O_CENTRAL],OPTION.SIBLINGS)
  \fi
  grps.print_person('\luaescapestring{#2}',\grps@option@depth,1,options,1)
}}
```

3.4 Lists

```
%%
```

Env `grampslisti`

```
\newenvironment{grampslisti}{%
\begin{list}{}%
{%
    \renewcommand{\makelabel}[1]{\bf\it\makebox{##1}}
    \setlength{\itemindent}{5ex}%
    \setlength{\leftmargin}{0pt}%
      \setlength{\labelwidth}{7ex}%
      \addtolength{\topsep}{-0.5\parskip}%
      %\listparindent \normalparindent
      \setlength{\parsep}{\parskip}%
    \setlength{\labelsep}{0pt}%
    \setlength{\itemsep}{0pt}%
    \setlength{\leftmargin}{2ex}%
    \setlength{\rightmargin}{0pt}%
    }%
}{\end{list}}
```

Env `grampslistii`

```
\newenvironment{grampslistii}{%
\begin{list}{}%
{%
    \renewcommand{\makelabel}[1]{\bf\it\makebox{##1}}
    \setlength{\itemindent}{5ex}%
```

```
    \setlength{\leftmargin}{0pt}%
      \setlength{\labelwidth}{7ex}%
      \addtolength{\topsep}{-0.5\parskip}%
      %\listparindent \normalparindent
      \setlength{\parsep}{\parskip}%
    \setlength{\labelsep}{0pt}%
    \setlength{\itemsep}{0pt}%
    \setlength{\leftmargin}{5ex}%
    \setlength{\rightmargin}{0pt}%
    }%
    \small
}{\end{list}}
```

Env grampslisti

```
\renewenvironment{grampslisti}%
{\begin{list}{}{%
  \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
  }}%
{\end{list}}
```

Env grampslistii

```
\renewenvironment{grampslistii}%
{\begin{list}{}{%

  \renewcommand{\makelabel}[1]{\bf\makebox{\makeromannumber{##1}\relax}}%
  }\small}
{\end{list}}
```

Env grampslistiii

```
\newenvironment{grampslistiii}
{\begin{list}{}{%
  \renewcommand{\makelabel}[1]{\makebox{\makealphanumber{##1}\relax}}%
  }\footnotesize}
{\end{list}}
```

Env grampslistiiii

```
\newenvironment{grampslistiiii}{\begin{list}{}{%
  \renewcommand{\makelabel}[1]{\makebox{\makeromannumber{##1}\relax}}}\footnotesize}{\end{list}}
```

Env grampslistiiiii

```
\newenvironment{grampslistiiiii}{\begin{list}{}{%
  \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
  }\small}{\end{list}}
```

Env grampslistiiiiii

```
\newenvironment{grampslistiiiiii}{\begin{list}{}{%
  \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
  }\footnotesize}{\end{list}}
```

Env grampslistiiiiiii

```
\newenvironment{grampslistiiiiiii}{\begin{list}{}{%
  \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
  }\footnotesize}{\end{list}}
```

Env grampslistiiiiiiii

```
\newenvironment{grampslistiiiiiiii}{\begin{list}{}{%
  \renewcommand{\makelabel}[1]{\bf\makebox{##1}}%
  }\tiny}{\end{list}}
```

Env compactitemize

```
\newenvironment{compactitemize}{%
\setlength\parskip{0pt}
\begin{itemize}[
  topsep=0pt,
  partopsep=0pt,
  parsep=0pt,
  itemsep=0pt,
  leftmargin=3ex,
  rightmargin=0pt,
  listparindent=0pt,
  labelwidth=3ex,
  labelsep=0pt,
  itemindent=0ex,
  align=left,
  after=\vspace{0pt}%\vspace{-\baselineskip}
  ]}{\end{itemize}%
  }%
```

3.5 Utils

```
%%
```

\modulo {⟨*value*⟩}modulo

```
\def\modulo#1#2{\number\numexpr\moduloop{#1}{#2}\relax}

 \def\truncdiv#1#2{((#1-(#2-1)/2)/#2)}
 \def\moduloop#1#2{(#1-\truncdiv{#1}{#2}*#2)}
```

\makeromannumber

```
\newcommand{\makeromannumber}[1]{\setcounter{result}{0}%
\addtocounter{result}{#1}%
\roman{result}\relax%
}
```

\makealphanumber

```
\newcommand{\makealphanumber}[1]{\setcounter{result}{0}%
\addtocounter{result}{#1}%
\alph{result}\relax%
}
```

3.6 Ifs

```
\newif\if@grps@option@shortperson@\@grps@option@shortperson@false%
\newif\if@grps@option@tree@\@grps@option@tree@false%
\newif\if@grps@option@parttree@\@grps@option@parttree@false%
\newif\if@grps@adobe@out@\@grps@adobe@out@false
\newif\if@grps@default@\@grps@default@false
\newif\if@grps@fullname@\@grps@fullname@true
\newif\if@grps@option@siblings@\@grps@option@siblings@false
```

Column and list handling

```
\newif\if@grps@mode@list@ \@grps@mode@list@false%
\newif\if@columnmode \@columnmodefalse%
```

`\startmulticolumn`

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

`\stopmulticolumn`

```
\newcommand{\stopmulticolumn}{%
    \if@columnmode\end{multicols}\@columnmodefalse\fi%
}%
```

`\startlist`

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

`\stoplist`

```
\newcommand{\stoplist}{\if@grps@mode@list@\end{grampslisti}\@grps@mode@list@false\fi}%

\newcounter{grps@itemize@depth}%
\setcounter{grps@itemize@depth}{0}%
```

`\grpsItemizeActualdepth`

```
\def\grpsItemizeActualdepth{1}
```

`\startitemize`

```
\newcommand{\startitemize}{%
    \ifthenelse{\numexpr\value{grps@itemize@depth} <\grpsItemizeActualdepth}%
    {\stepcounter{grps@itemize@depth}\begin{compactitemize}}{}%
}%
```

\stopitemize

```
\newcommand{\stopitemize}[1]{%
    \def\grpsItemizeActualdepth{#1}%
    \whiledo{\numexpr\value{grps@itemize@depth} =\grpsItemizeActualdepth}%
        {\end{compactitemize}\addtocounter{grps@itemize@depth}{-1}}%
    \edef\grpsItemizeActualdepth{\arabic{grps@itemize@depth}}%
}%
```

\doitem

```
\newcommand{\doitem}[2][$\cdot$]{%
    \ifthenelse{\grps@option@itemize > 0}{
    \ifthenelse{\grps@option@itemize < \grpsItemizeActualdepth}{#2}%
    {\startitemize\item[#1]}}{}%
}%
```

3.7 Counters

```
\newcounter{Generation}
\newcounter{grpsNumber}
\newcounter{NewGeneration}
\newcounter{result}
```

3.8 Keys (grpskeys)

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

gen amount of generations or “all”

```
\def\grps@gen@all{all}
\define@key{grpskeys}{gen}{\def\grps@key@temp{#1}\ifx\grps@gen@all\grps@key@temp%
    \def\generations{1000}\loadgramps\directlua{grps.generations = 1000}%
    \else%
    \def\generations{#1}\loadgramps\directlua{grps.generations = #1}%
    \fi}%
```

depth generation depth by (grand)children

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

genpiece cut into pieces of n generations

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

col column settings

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

show siblings

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

show tree

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

parttree

```
\define@key{grpskeys}{parttree}[true]{%
   \def\partialbool{#1}%
   \ifx\partialbool\stringtrue\@grps@option@parttree@true\else\@grps@option@parttree@false\fi%
   }%
```

abobe output

```
\define@key{grpskeys}{adobe}[true]{%
   \def\partialbool{#1}%
   \ifx\partialbool\stringtrue\@grps@adobe@out@true\else\@grps@adobe@out@false\fi%
   }%
\define@key{grpskeys}{noadobe}[true]{\@grps@adobe@out@false}%
```

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

```
\xdef\grps@option@long{4}
\define@key{grpskeys}{long}[4]{\xdef\grps@option@long{#1}\@grps@option@shortperson@false}%
\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)

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

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

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

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

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

option default start with default options

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

option fullname sets show fullname

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

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

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

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

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

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

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

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

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

option itemize itemize information

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

3.9 Date

```
%%
```

`\getshortdate` {⟨*yyyy-m-d*⟩}
`\getshortdate`{2024-11-1}:
default: 11-1-2024
dutch: 1-11-2024

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

```
  \iflanguage{dutch}{%
  \ifx\grpsDyear\empty{}\else%
  \ifx\grpsDmonth\empty{}\else%
  \ifx\grpsDday\empty{}\else%
  \grpsDday-\fi\grpsDmonth-\fi\grpsDyear\fi
  }{%
  \ifx\grpsDyear\empty{}\else%
  \ifx\grpsDmonth\empty{}\else\grpsDmonth-%
  \ifx\grpsDday\empty{}\else%
  \grpsDday-\fi\fi\grpsDyear\fi%
  }%
}
```

`\getlongdate` {⟨*yyyy-m-d*⟩}
`\getlongdate{2024-11-1}`:
default: November 1 2024
dutch: 1 November 2024

```
\newcommand{\getlongdate}[1]{%
 \def\grpsDyear{}\def\grpsDmonth{}\def\grpsDday{}%
 \StrCut[1]{#1}{-}\grpsDyear\grpsDtemp%
 \StrCut[1]{\grpsDtemp}{-}\grpsDmonth\grpsDday%
 \iflanguage{dutch}{%
 \ifx\grpsDyear\empty{}\else%
 \ifx\grpsDmonth\empty{}\else%
 \ifx\grpsDday\empty{}\else%
 \grpsDday\,\fi\grpsTmonth{\grpsDmonth} \fi\grpsDyear\fi
 }{%
 \ifx\grpsDyear\empty{}\else%
 \ifx\grpsDmonth\empty{}\else\grpsTmonth{\grpsDmonth} %
 \ifx\grpsDday\empty{}\else%
 \grpsDday\ \fi\fi\grpsDyear\fi%
 }%
}
```

3.10 Event

```
%%
```

`\grpsIresidence`

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

`\grpsIeducation`

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

`\grpsIoccupation`

```
\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)

```
\newcommand{\grpsEvent}[9]{%
\if@grps@option@shortperson@%
\doitem[\getshorteventtype{#1}]{\getshorteventtype{#1}}%
\ifcase#3{}\or{$<$}\or{$>$}\or{$\sim$}\or{}\or{}\or{}\else{}\fi
\ifthenelse{#3 < 6}{%
\ifx&#5&\relax\else\,\getshortdate{#5}\fi%
\ifthenelse{#3 > 3}{\,-\,\ifx&#5&\relax\else\getshortdate{#7}\fi}{}%
\ifx&#6&\relax\else{:#6}\fi}%
{}%text only%
\else% Long text person
\doitem[\makehijzij{\grpssex}]{}%
\getlongeventtype{#1} %
\ifcase#3{\grpsTop}\or{\grpsTvoor}\or{\grpsTna}\or{\grpsTrond}%
    \or{\grpsTvan}\or{\grpsTtussen}\or{}\else{}\fi%
\ifthenelse{#3 < 6}{%
\ifx&#5&{leeg}\else\getlongdate{#5}\fi%
\ifthenelse{#3 > 3}{ \grpsTtot\ifx&#7&\relax\else\getlongdate{#7}\fi}{}%
\ifx&#6&\relax\else\ \grpsTte\,#6\fi%
\typeout{#9}%
\ifx&#9&\relax\else\ (#9)\fi%
}%
{#9}%text only
\fi%
}
```

\getshorteventtype TYPE

```
\newcommand{\getshorteventtype}[1]{
\ifcase#1{CUSTOM}% = 0,
\or{\gtrsymMarried}%    MARRIAGE = 1,
\or{MARR\_SETTL}% = 2,
\or{MARR\_LIC}% = 3,
\or{MARR\_CONTR}% = 4,
\or{MARR\_BANNS}% = 5,
\or{\gtrsymEngaged}%    ENGAGEMENT = 6,
\or{\gtrsymDivorced}%    DIVORCE = 7,
\or{DIV\_FILING}% = 8,
\or{ANNULMENT}% = 9,
\or{\gtrsymPartnership}%    MARR\_ALT = 10,
\or{ADOPT}% = 11,
\or{\gtrsymBorn}%    BIRTH = 12,
\or{\gtrsymDied}%    DEATH = 13,
\or{ADULT\_CHRISTEN}% = 14,
\or{\gtrsymBaptized}%    BAPTISM = 15,
\or{BAR\_MITZVAH}% = 16,
\or{BAS\_MITZVAH}% = 17,
\or{BLESS}% = 18,
\or{\gtrsymBuried}%    BURIAL = 19,
\or{CAUSE\_DEATH}% = 20,
\or{CENSUS}% = 21,
\or{CHRISTEN}% = 22,
\or{CONFIRMATION}% = 23,
```

```
\or{CREMATION}% = 24,
\or{DEGREE}% = 25,
\or{\grpsIeducation}%    EDUCATION = 26,
\or{ELECTED}% = 27,
\or{EMIGRATION}% = 28,
\or{FIRST\_COMMUN}% = 29,
\or{IMMIGRATION}% = 30,
\or{GRADUATION}% = 31,
\or{MED\_INFO}% = 32,
\or{MILITARY\_SERV}% = 33,
\or{NATURALIZATION}% = 34,
\or{NOB\_TITLE}% = 35,
\or{NUM\_MARRIAGES}% = 36,
\or{\grpsIoccupation}%    OCCUPATION = 37,
\or{ORDINATION}% = 38,
\or{PROBATE}% = 39,
\or{PROPERTY}% = 40,
\or{RELIGION}% = 41,
\or{\grpsIresidence}%    RESIDENCE = 42,
\or{RETIREMENT}% = 43,
\or{WILL}% = 44
\else{UNKNOWN (#1)}\fi%
}
```

\getlongeventtype

```
\newcommand{\getlongeventtype}[1]{
\ifcase#1{CUSTOM}% = 0,
\or{\grpsTtrouwde}%    MARRIAGE = 1,
\or{MARR\_SETTL}% = 2,
\or{MARR\_LIC}% = 3,
\or{MARR\_CONTR}% = 4,
\or{MARR\_BANNS}% = 5,
\or{\grpsTverloofde}% = 6,
\or{\grpsTscheidde}%    DIVORCE}% = 7,
\or{DIV\_FILING}% = 8,
\or{ANNULMENT}% = 9,
\or{MARR\_ALT}% = 10,
\or{ADOPT}% = 11,
\or{\grpsTgeboren}%    BIRTH}% = 12,
\or{\grpsToverleed}%    DEATH}% = 13,
\or{ADULT\_CHRISTEN}% = 14,
\or{\grpsTgedoopt}%    BAPTISM}% = 15,
\or{BAR\_MITZVAH}% = 16,
\or{BAS\_MITZVAH}% = 17,
\or{BLESS}% = 18,
\or{\grpsTbegraven}%    BURIAL}% = 19,
\or{CAUSE\_DEATH}% = 20,
\or{CENSUS}% = 21,
\or{CHRISTEN}% = 22,
\or{CONFIRMATION}% = 23,
\or{CREMATION}% = 24,
\or{DEGREE}% = 25,
\or{\grpsTopleiding}%    EDUCATION}% = 26,
\or{ELECTED}% = 27,
\or{EMIGRATION}% = 28,
\or{FIRST\_COMMUN}% = 29,
```

```
\or{IMMIGRATION}% = 30,
\or{GRADUATION}% = 31,
\or{MED\_INFO}% = 32,
\or{MILITARY\_SERV}% = 33,
\or{NATURALIZATION}% = 34,
\or{NOB\_TITLE}% = 35,
\or{NUM\_MARRIAGES}% = 36,
\or{\grpsTberoepis}%    OCCUPATION}% = 37,
\or{ORDINATION}% = 38,
\or{PROBATE}% = 39,
\or{PROPERTY}% = 40,
\or{RELIGION}% = 41,
\or{\grpsTwoonde}%    RESIDENCE}% = 42,
\or{RETIREMENT}% = 43,
\or{WILL}% = 44
\else{UNKNOWN}\fi
}
```

3.11 Ancetsor

```
 %%
 \def\gtrkv@treeprefix{TREE}
 \def\gtrkv@treesuffix{}
 \tikzset{
/gtr/id/.code={%
\xdef\gtr@gkv@idlink{#1}%
\xdef\gtr@gkv@id{\expandonce\gtrkv@idprefix\unexpanded{#1}\expandonce\gtrkv@idsuffix}},
/gtr/tree prefix/.store in=\gtrkv@treeprefix,
/gtr/tree suffix/.store in=\gtrkv@treesuffix,
/gtr/thistree/.code  ={\xdef\gtr@gkv@thistree{\expandonce\gtrkv@treeprefix\unexpanded{#1}\expando
/gtr/parenttree/.code={\xdef\gtr@gkv@parenttree{\expandonce\gtrkv@treeprefix\unexpanded{#1}\expan
/gtr/childtree/.code ={\xdef\gtr@gkv@childtree{\expandonce\gtrkv@treeprefix\unexpanded{#1}\expand
/gtr/lefttree/.code={\xdef\gtr@gkv@lefttree{\expandonce\gtrkv@treeprefix\unexpanded{#1}\expandonc
/gtr/righttree/.code ={\xdef\gtr@gkv@righttree{\expandonce\gtrkv@treeprefix\unexpanded{#1}\expand
/gtr/haschildren/.style = {%
name code  = {%
\hyperlink{\gtr@gkv@idlink}{\gtrPrintName@full (\gtrDBkekule)}%
%\ifdefined\gtr@gkv@lefttree\showocg{\gtr@gkv@lefttree}{$\leftnarrow$}\fi
%\gtr@gkv@id
},
},
    /gtr/hasparents/.style = {%
name code  ={%
\if@grps@adobe@out@%
\showocg{\gtr@gkv@parenttree}{\fbox{$\uparrow$}}\else%
\hyperlink{GRAPH\expandonce\gtr@gkv@parenttree}{\fbox{$\uparrow$}}\fi%
\\\vspace{0mm}%
\hyperlink{\gtr@gkv@idlink}{\gtrPrintName@full (\gtrDBkekule)}%
},
},
 }
```

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

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

```
function grps_prepare_trees(base,starts,piece)
for i, h in ipairs(starts) do
local filebase = 'data.'..base_id..'.kwartierstaat.'..gen_piece.."."..i
local filename = kpse.find_file(filebase)
if filename == nil then
tex.print("\\typeout{create file "..filebase.."}")
local file = io.open(filebase, "w")
if file then
file:write(grps.ancestor_tree_data(h,gen_piece))
file:close()
else
tex.print("\\typeout{Error: Could not open the file.}")
end
end
end
end
\end{luacode*}
```

`\AncestorTree` `{`⟨*person_handle or person_id*⟩`}` `{`⟨*maximum gennerations*⟩`}`

`\AncestorTree{I0018}` = `[`⟨*gen*⟩`]` total amount of generations (default=3)

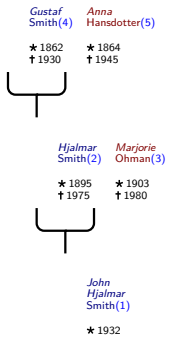

Figure 1: Ancestors of John Hjalmar Smith

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

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

```
tex.print("\\node[] (TREEBASE"..base_id..") at (0,0) {};")
for i, h in ipairs(tree_starts) do
tex.print("\\AncestorTreePart{"..base_id.."}{\\generationpiece}{"..i.."}{"..h.."}")
end
tex.print("\\end{tikzpicture}\\caption{\\label{GRAPHTREE#2}\grpsTkwartierstaatvan{#2}}\\end{cente
else
for i, h in ipairs(tree_starts) do
tex.print("\\begin{figure}[htb]\\begin{center}\\hypertarget{GRAPHTREE"..h.."}{}\\begin{tikzpictur
tex.print("\\node[] (TREEBASE"..base_id..") at (0,0) {};")
tex.print("\\AncestorTreePart{"..base_id.."}{\\generationpiece}{"..i.."}{"..h.."}")
tex.print("\\end{tikzpicture}")
tex.print("\\caption{\\grpsTkwartierstaatvan{\\fullname{"..h.."}}}")
tex.print("\\end{center}\\end{figure}")
end
end
}
}
```

`\AncestorTreePart` {⟨*base id*⟩}
{⟨*generation piece*⟩}
{⟨*folowup number*⟩} start with 1 which is at start visible
{⟨*id parent part tree*⟩}

```
\newcommand{\AncestorTreePart}[4]{
\if@grps@adobe@out@%
\ifthenelse{#3 = 1}{\def\visibility{on}}{\def\visibility{off}}
\else%
\def\visibility{on}%
\fi%
\typeout{\visibility, #1, #2, #3, #4}
\let\gtr@gkv@childtree\undefined
\let\gtr@gkv@lefttree\undefined
\let\gtr@gkv@righttree\undefined
\begin{scope}[ocg={
name={Tree #3},
ref=TREE#4,
visibility=\visibility,
opts={radiobtngrp=myRadioButtons}
}]%
\genealogytree[%
set position = #4#4 at TREEBASE#1,
template=database traditional,
level size=1.2cm,
level #2-1/.style={level size = 1.7cm},
%level 0/.style={level size = 1.7cm},
node size=\page@piece@part\columnwidth,%2.2
id prefix = #4,
date format=yyyy,
list separators={\par}{ }{}{},
    name font=\gtrifmale{\selectfont\color{blue!50!black}}{\selectfont\color{red!50!black}},
    name code  ={\hyperlink{\gtr@gkv@idlink}{\gtrPrintName@full (\gtrDBkekule)}},
]{input{data.#1.kwartierstaat.#2.#3}}
\if@grps@adobe@out@%
\ifdefined\gtr@gkv@childtree\node[draw,blue] (c) at (#4#4.south) {\showocg{\gtr@gkv@childtree}{$\
\ifdefined\gtr@gkv@lefttree\node [draw,blue] (a) at (#4#4.west) {\showocg{\gtr@gkv@lefttree}{$\le
\ifdefined\gtr@gkv@righttree\node[draw,blue] (b) at (#4#4.east) {\showocg{\gtr@gkv@righttree}{$\r
```

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

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

```
\newcommand{\Kwartierstaat}[2][]{%
\loadgramps%
\typeout{Kwartierstaat van #2}%
\setkeys{grpskeys}{gen=2,
col=2,
genpiece=200,
fullname,
depth=2,
relations=3,
long,
media=0,
tree=false,
lifeevents=3,
parents=2,
#1}
\typeout{Start kwartier = \generations, \generationpiece}
\directlua{grps.set('\luaescapestring{#2}',\luaescapestring{\generations})}%
\setcounter{Generation}{1}%
\setcounter{NewGeneration}{1}%

\def\grps@kekule{1}%
\ifthenelse{\generations > \generationpiece}%
{\@grps@option@parttree@true}%
{\edef\generationpiece{\generations}\@grps@option@parttree@false}%
```

Start the loop

```
    \@grps@mode@list@false\@columnmodefalse
\startmulticolumn
\directlua{
    base_id = "\luaescapestring{#2}"
    new_generation = 2
    generation = 2
    local max_gen = \luaescapestring{\generations}
    gen_piece = \luaescapestring{\generationpiece}
    local tree_starts = grps.set_ancestor_tree(base_id,max_gen,gen_piece)
local deel_kwartieren = grps.set_kwartier_pieces(tree_starts,gen_piece)
\if@grps@option@tree@ grps_prepare_trees(base_id,tree_starts,gen_piece) \fi
\if@grps@option@parttree@%
token.set_macro("page@piece@part",grps.format(".3f", (1/(2^(gen_piece-1)))-0.04 ),"global")
\else
token.set_macro("page@piece@part",grps.format(".3f", (1/(2^(max_gen-1)))-0.04 ),"global")
\fi
```

```
for i, h in ipairs(tree_starts) do
%tex.print("\\typeout{stuk "..i.."}")
for j,k in pairs(deel_kwartieren[i]) do
local handle=grps.kekule_id(k)
%tex.print("\\typeout{stuk "..i.." - "..k.." "..handle.."}")
if j==1 then
\if@grps@option@tree@ %
tex.print("\\stoplist\\stopmulticolumn")
tex.print("\\begin{figure}[htb]\\begin{center}")
tex.print("\\hypertarget{GRAPHTREE"..handle.."}{\\grpsTkwartierstaatvan{\\fullname{"..handle.."}}
tex.print("\\begin{tikzpicture}")
tex.print("\\node[] (TREEBASE"..base_id..") at (0,0) {};")
tex.print("\\AncestorTreePart{"..base_id.."}{\\generationpiece}{"..i.."}{"..handle.."}")
tex.print("\\end{tikzpicture}")
tex.print("\\end{center}\\end{figure}")
\fi
tex.print("\\startmulticolumn")
tex.print("\\startlist")
tex.print("\\item["..k.."]")
tex.print("\\hyperlink{"..handle.."}{\\fullname{"..handle.."}}")
\if@grps@option@parttree@%
tex.print("\\addcontentsline{toc}{subsection}{\\grpsTkwartierstaatvan{\\fullname{"..handle.."}}("
\fi%
else
\if@grps@option@parttree@%
\else%
if k >= new_generation then
tex.print("\\stoplist")
tex.print("\\subsection*{\\grpsTvoorouder{"..generation.."}}")
tex.print("\\addcontentsline{toc}{subsection}{\\grpsTvoorouder{"..generation.."}}")
new_generation = new_generation*2
generation  = generation+1
end
\fi%
tex.print("\\startlist")
tex.print("\\item["..k.."]")
print("Preparing person "..k)
tex.print("\\grpsPrintPerson[]{"..handle.."}")%
end
end
end
tex.print("\\stoplist\\stopmulticolumn")
}
}
```

3.12 Descendence

```
%%
```

\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}

```
\newcommand{\Descendence}[2][]{%
    \loadgramps%
    \typeout{Nageslacht van #2}%
```

```
    \setkeys{grpskeys}{gen=2,depth=1,col=2,short,#1}
    \directlua{%
      grps.set_descendence('\luaescapestring{#2}',\luaescapestring{\generations},true)}%
    \setcounter{Generation}{1}%
    \setcounter{NewGeneration}{1}%
    \setcounter{grpsNumber}{1}%
    \@grps@mode@list@false%
    \@columnmodefalse%
```

Start the loop

```
    \startmulticolumn%
    \whiledo{ \numexpr\value{Generation} - 1 < \generations }{%
        \typeout{printing \arabic{grpsNumber}}%
        \directlua{
                local ghi = grps.get_descendent(\arabic{grpsNumber})
                tex.print("\\setcounter{Generation}{"..ghi[1].."}")
                tex.print("\\gdef\\handle{"..ghi[2].."}")
                tex.print("\\gdef\\itemindex{"..ghi[3].."}")
                tex.print("\\gdef\\grampsid{"..ghi[4].."}")
            }%
        \ifthenelse{\numexpr\value{Generation} -1 < \generations}{%
            \ifthenelse{\value{NewGeneration} = \value{Generation}}{%
                \subsection*{Generatie \arabic{Generation}}
                \addcontentsline{toc}{subsection}{Generation \arabic{Generation}}
                \stepcounter{NewGeneration}}{}%
        \startlist
        \item[\itemindex] \grpsFirstPerson{\handle}%
        \stoplist
        }{}
        \stepcounter{grpsNumber}%
    }%
    \stoplist%
    \stopmulticolumn%
    }
```

3.13 Descendence text per branche

```
%%
```

`\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}

```
\newcommand{\Descendenceb}[2][]{%
    \loadgramps%
    \typeout{Nageslacht van #2}%
    \setkeys{grpskeys}{gen=2,depth=1,col=2,genpiece=3,short,#1}
    \directlua{grps.set_descendence('\luaescapestring{#2}',\luaescapestring{\generations},false)}
    \setcounter{Generation}{1}%
    \setcounter{NewGeneration}{1}%
    \setcounter{grpsNumber}{1}%
    \@grps@mode@list@false%
    \@columnmodefalse%
    \startmulticolumn
```

Start the loop

```
    \whiledo{ \numexpr\value{Generation} - 1 < \generations }{%
        \directlua{
                local ghi = grps.get_descendent(\arabic{grpsNumber})
                tex.print("\\setcounter{Generation}{"..ghi[1].."}")
                tex.print("\\gdef\\handle{"..ghi[2].."}")
                tex.print("\\gdef\\itemindex{"..ghi[3].."}")
                tex.print("\\gdef\\grampsid{"..ghi[4].."}")
            }%
        \ifthenelse{\numexpr\value{Generation} -1 < \generations}{%
            \edef\modGeneration{\modulo{\arabic{Generation}}{\generationpiece}}% Generations Modul
            \typeout{modgeneration = \modGeneration, \arabic{Generation}}%
            \ifthenelse{\modGeneration = 1}{%
                \stoplist%
                \stopmulticolumn%
                \begin{center}
                \pgfmathtruncatemacro{\result}{\generationpiece +1}%
                \DescendentTree[parttree=true]{\grampsid}{\result}%
                \end{center}
            }{\typeout{modGeneration != 1}}%
        \startmulticolumn
        \startlist
        \item[\itemindex] \grpsFirstPerson{\handle}%
        }{}
        \stepcounter{grpsNumber}%
    }%
    \stoplist%
    \stopmulticolumn%
    }
```

Descendent Tree

```
%%
```

`\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}

```
\newcommand{\Descendenceall}[2][]{%
    \loadgramps%
    \typeout{Nageslacht van #2}%
    \setkeys{grpskeys}{gen=2,depth=1,short,genpiece=2,col=2,#1}
    \directlua{%
grps.set_descendence('\luaescapestring{#2}',\luaescapestring{\generations})}%
    \setcounter{Generation}{1}%
    \setcounter{NewGeneration}{1}%
    \setcounter{grpsNumber}{1}%
    \startmulticolumn
```

Start the loop

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

```
                tex.print("\\setcounter{Generation}{"..ghi[1].."}")
                tex.print("\\gdef\\handle{"..ghi[2].."}")
                tex.print("\\gdef\\itemindex{"..ghi[3].."}")
                tex.print("\\gdef\\grampsid{"..ghi[4].."}")
            }%
        \typeout{gerneration=\arabic{Generation}, \itemindex - \grampsid (\handle)}
        \ifthenelse{\numexpr\value{Generation} -1 < \generations}{%
            \edef\modGeneration{\modulo{\arabic{Generation}}{\generationpiece}}% Generations Modul
            \typeout{modgeneration = \modGeneration, \arabic{Generation}}%
            \ifthenelse{\value{NewGeneration} = \value{Generation}}{% NEXT GENERATION
                \subsection*{Generatie \arabic{Generation}}
                \addcontentsline{toc}{subsection}{Generation \arabic{Generation}}
                \stepcounter{NewGeneration}}{}%
            \ifthenelse{\modGeneration = 1}{%
                \stoplist%
                \stopmulticolumn%
                \begin{center}
                \pgfmathtruncatemacro{\result}{\generationpiece +1}%
                \DescendentTree[parttree=true]{\grampsid}{\result}%
                \end{center}
            }{\typeout{modGeneration != 1}}%
        }{}%
        \startmulticolumn%
        \startlist%
        \item[\itemindex] %\arabic{grpsNumber}. \itemindex-\grampsid %, generation=\arabic{Genera
        \grpsFirstPerson{\handle}%
        \stepcounter{grpsNumber}%
    }%
    \stoplist%
    \stopmulticolumn%
    }%
```

\DescendentTree

3.14 Descendent Tree

```
\newcommand\DescendentTree[3][]{
\loadgramps%
\typeout{#1,#2,#3}
\setkeys{grpskeys}{parttree=true,#1}
\directlua{
    local id = "\luaescapestring{#2}"
    local max_gen = \luaescapestring{#3}
    local command = "\\DescendentTree{"..id.."}{"..max_gen.."}"
    filebase = 'data.'..id..'.descendent.'..max_gen
%%    io.write("\string\n In "..command.." gebruikt "..filebase)
    local filename = kpse.find_file(filebase)
%%    if filename == nil then
        io.write("\string\n In "..command.." creating "..filebase)
        local file = io.open(filebase, "w")
        if file then
            file:write(grps.descendent_tree_data(id,max_gen))
            file:close()
            filename=filebase
        else
            io.write("\string\n In "..command.." ERROR "..filename.." not OPENED")
            tex.print("Error: Could not open the file.")
```

```
        end
```

end

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

\if@grps@option@parttree@
\typeout{parttree true}
\tikzset{%
    /gtr/haschildren/.style={name code  ={\hypertarget{CHILD\gtr@gkv@id}{}\hyperlink{STAM\gtr@gkv
    /gtr/hasparents/.style={name code  ={\hypertarget{STAM\gtr@gkv@id}{}\hyperlink{CHILD\gtr@gkv@
    }%
\else
\typeout{parttree false}
\tikzset{%
    /gtr/haschildren/.style={name code  ={\gtrPrintName@full}},%
    /gtr/hasparents/.style={name code  ={\gtrPrintName@full}},%
    }%
\fi
\begin{tikzpicture}
\genealogytree[%
    template=database traditional,
    level size=0.18\columnwidth,
    level distance = 7mm,
    node size=6mm,%2.2
    child distance=-1.5mm,
    parent distance=-2.5mm,
    date format=yyyy,
    timeflow=right,
    list separators={\newline}{ }{}{},
    name font=\gtrifmale{\selectfont\color{blue!50!black}}{\selectfont\color{red!50!black}},
    name code  ={\hyperlink{\gtr@gkv@id}{\gtrPrintName@full}},
    childless/.style={
        family={
            edges={foreground={red!20!black,line width=0.1mm},xshift=-1mm},
            },
        },
]{input{data.#2.descendent.#3}}
\end{tikzpicture}
}
```

3.15 Languages

English

```
\def\grpsThij{he }%
\def\grpsTHij{He }%
\def\grpsTzij{she }%
\def\grpsTZij{She }%
\def\grpsTis{is }%
\def\grpsTmet{with }%
\def\grpsTen{and }%
\def\grpsTZoon{Son of }%
\def\grpsTDochter{Daughter of }%
```

used in events

```
\def\grpsTop{at }
\def\grpsTvoor{before }
\def\grpsTna{after }
\def\grpsTrond{about }
\def\grpsTvan{from }
\def\grpsTtussen{between }
\def\grpsTtrouwde{married }
\def\grpsTscheidde{divorced }
\def\grpsTverloofde{engaged }
\def\grpsTkoppelde{had a relation }
\def\grpsTgeboren{is born }
\def\grpsToverleed{died }
\def\grpsTbegraven{is buried}
\def\grpsTgedoopt{is baptiezed }
\def\grpsTwoonde{lived }
\def\grpsTte{in }
\def\grpsTtot{til}
\def\grpsTberoepis{occupation is }
\def\grpsTopleiding{is educationed }
\newcommand{\grpsTmonth}[1]{%
 \ifcase#1
   \or January\or February\or March\or April\or May\or June%
   \or July\or August\or September\or October\or November\or December%
 \else
   Invalid month number
 \fi}%
\def\grpsTkwartierstaatvan#1{Ancestors of #1}%
\def\grpsTdescendenceof#1{Descendence of #1}%
\def\grpsTvoorouder#1{Generation #1}%
\newcommand{\makehijzij}[1]{\ifcase#1{zij }\or{hij }\else oo\fi}%
```

Dutch

```
\iflanguage{dutch}{%
\def\grpsThij{hij }%
\def\grpsTHij{Hij }%
\def\grpsTzij{zij }%
\def\grpsTZij{Zij }%
\def\grpsTmet{met }%
\def\grpsTen{en }%
\def\grpsTZoon{Zoon van }%
\def\grpsTDochter{Dochter van }%
```

used in events

```
\def\grpsTop{op }
\def\grpsTvoor{voor }
\def\grpsTna{na }
\def\grpsTrond{rond }
\def\grpsTvan{van }
\def\grpsTtot{tot }
\def\grpsTtussen{tussen }
\def\grpsTtrouwde{trouwde }
\def\grpsTscheidde{scheidde }
```

```
\def\grpsTkoppelde{had een relatie }
\def\grpsTverloofde{verloofde }
\def\grpsTgeboren{is geboren }
\def\grpsToverleed{is overleed }
\def\grpsTbegraven{is begraven }
\def\grpsTgedoopt{is gedoopt }
\def\grpsTwoonde{woonde }
\def\grpsTte{in }
\def\grpsTberoepis{beroep is }
\def\grpsTopleiding{is opgeleid }
\renewcommand{\grpsTmonth}[1]{%
 \ifcase#1
   \or januarie\or februarie\or maart\or april\or mei\or juni%
   \or juli\or augustus\or september\or oktober\or november\or december%
 \else
   incorrect maand numbmer
 \fi}%
\def\grpsTkwartierstaatvan#1{Kwartierstaat van #1}
\def\grpsTdescendenceof#1{Afstammelingen van #1}%
\renewcommand\grpsTvoorouder[1]{%
 \ifcase#1%
   \or Kwartierdrager%
       \or Ouders \or Grootouders \or Overgrootouders \or Betovergrootouders%
       \or Oudouders \or Oudgrootouders\or Oudovergrootouders \or Oudbetovergrootouders%

       \or Stamouders \or Stamgrootouders \or Stamovergrootouders \or Stambetovergrootouders%
       \or Stamoudouders \or Stamoudgrootouders\or Stamoudovergrootouders \or Stamoudbetovergroo

       \or Edelouders \or Edelgrootouders \or Edelovergrootouders \or Edelbetovergrootouders%
       \or Edeloudouders \or Edeloudgrootouders\or Edeloudovergrootouders \or Edeloudbetovergroo

       \or Edelstamouders \or Edelstamgrootouders \or Edelstamovergrootouders \or Edelstambetove
       \or Edelstamoudouders \or Edelstamoudgrootouders\or Edelstamoudovergrootouders \or Edelst

       \or Voorouders \or Voorgrootouders \or Voorovergrootouders \or Voorbetovergrootouders%
       \or Vooroudouders \or Vooroudgrootouders\or Vooroudovergrootouders \or Vooroudbetovergroo

       \or Voorstamouders \or Voorstamgrootouders \or Voorstamovergrootouders \or Voorstambetove
       \or Voorstamoudouders \or Voorstamoudgrootouders\or Voorstamoudovergrootouders \or Voorst

       \or Vooredelouders \or Vooredelgrootouders \or Vooredelovergrootouders \or Vooredelbetove
       \or Vooredeloudouders \or Vooredeloudgrootouders\or Vooredeloudovergrootouders \or Voored

       \or Vooredelstamouders \or Vooredelstamgrootouders \or Vooredelstamovergrootouders \or Vo
       \or Vooredelstamoudouders \or Vooredelstamoudgrootouders\or Vooredelstamoudovergrootouder
 \else%
   Generatie #1%
 \fi}%
}{}%
```