OP-Bible – Technical Documentation

The code of the `op-bible.opm` macro file is described here.

```
3  \_codedecl \processbooks {OpBible: macros for creating annotated Bible}
```

1   Preparatory work

Loading packages.

```
12  \_load[vlna]   % single-letter prepositions and splitting hyphen managed specially in Czech
13  \_load[mte]    % micro typographical extensions
14
15  \_namespace{opb}
```

Basic settings of TEX parameters.

```
21  \_newdimen\.lrmargin  \.lrmargin=10mm
22  \_margins/2 a4 (23,27,20,20)mm
23
24  \_typosize[11/13]        % typesetting size of Bible text
25  \_hyperlinks\Blue\Blue  % hyperlinks activated
26
27  \_parindent=20pt
28  \_nopagenumbers
29  \_mte_enablemte          % micro typographical extensions enabled
30  \_vlna_singlechars {Czech}{AaIiVvOoUuSsZzKk}  % lowercase "a" added to this family
31
32  \_showboxbreadth=0
33  \_let\notecolor=\Red
34
35  \_def\LightGrey {\_setcmykcolor{0 0 0 .1}}
36  \_def\LiRed {\_setcmykcolor{0 .2 .2 0}}
```

Fonts.

```
42  \_fontfam[lm]
43  \_fontfam[Heros]        % fonts for notes
44  \_isfile{f-biblon.opm}\_iftrue
45    \_fontfam[biblon]     % fonts for Bible text
46  \_else
47    \_let\Biblon=\LMfonts
48  \_fi
49
50  \_fontdef\.bookfont{\_setfontsize{at19.pt}\_bf}
51  \_fontdef\.chapfont{\_setfontsize{at13.pt}\_bf}
52  \_fontdef\.markfont{\_setfontsize{at7pt}\_rm}
53  \_fontdef\.captionfont{\Heros\cond\_setfontsize{at8pt}\_bf}
54  \_def\.headfont{\.Biblon\_setfontsize{at10pt}\_rm}
55  \_nsprivate \Biblon ;
```

Auxiliary macros. `\.printwarn` {⟨*text*⟩} prints warning. `\.sedef` {⟨*name*⟩}{⟨*body*⟩} is expanded `\sdef`. `\.myaddto` {⟨*macro-name*⟩}{⟨*text*⟩} adds ⟨*text*⟩ to \⟨*macro-name*⟩ globally. Moeower it defines the undefined macro by `\sdef{`⟨*macro-name*⟩`}{`⟨*text*⟩`}`.

```
65  \_let\.printwarn=\_opwarning
66  \_def \.sedef #1{\_ea\_edef \_csname#1\_endcsname}
67  \_long\_def\.myaddto#1#2{\_ifcsname#1\_endcsname
68    \_gobal\_ea\_addto\_csname#1\_endcsname{#2}\_else \_global\_sdef{#1}{#2}\_fi}
```

We prepare expandable if-macros:
`\.isspacein` ⟨*text*⟩ `\_iftrue` is true if ⟨*text*⟩ includes a space.

`\.iscolonin` ⟨*text*⟩`:\_iftrue` is true if ⟨*text*⟩ includes a colon.
`\.isdivisin` ⟨*text*⟩`-\_iftrue` is true if ⟨*text*⟩ includes a divis.

```
77  \_def\.isspacein #1 #2\_iftrue{\_isempty{#2}\_iffalse}
78  \_def\.iscolonin #1:#2\_iftrue{\_isempty{#2}\_iffalse}
79  \_def\.isdivisin #1-#2\_iftrue{\_isempty{#2}\_iffalse}
```

2   The main loop over Bible books

The `\processbooks` macro does two loops over all marks in `\printedbooks`. The macro `\printedbooks` is a list of ⟨*a-marks*⟩ of Bible books separated by spaces and it must be defined in the main file. The `\_useit` trick is used here in order we want to add ⟨*space*⟩`{}` at the end of the expanded `\printedbooks`. The first loop body sets `\pbook!`⟨*a-mark*⟩ used for hyperlinks. The second loop body does:

- Defines `\amark` as ⟨*a-mark*⟩ (an actual mark of the book used in the text).
- Defines `\bmark` as ⟨*b-mark*⟩ (a mark of the book used in file names).
- Defines `\.btit` as the book title.
- Saves ⟨*a-mark*⟩ to the `\.currbook` macro.
- Calls `\.newbook{`⟨*a-mark*⟩`}`
- Prints title of the book to the terminal and to the log.
- Calls `\bex!<a-mark>` in order to apply the `\BookException` data.
- Inputs introduction file if it exists. The real `\input` and formatin of the introduction text is done by the `\.printintro` macro.
- Inputs format definition file if it exists. Information is saved to the TeX memory.
- Inputs notes file if it exists. The notes are saved to the TeX memory.
- Calls `\bpr!<a-mark>` in order to apply the `\BookPre` data.
- Inputs `txs` file with original text of the Bible using `\.bibleinput`, i.e. prints the text from `txs` file with notes from the TeX memory.
- Calls `\bpo!<a-mark>` in order to apply `\BookPost` data.

Note that the macros `\introfile`, `\fmtfile`, and `\notesfile` give the location of aprropriate files and these macros must be defined by the user in the main file.

Note2: each book of the Bible is processed in the group. It means that all data from notes, formats etc. are stored in the memory only temporary for processing single book. After the Book is finalized, the TeX memory is freed.

```
119  \_def\.processbooks {\_par
120     \.checknochapbooks
121     \_useit{\_ea\.processbooksA \printedbooks} {}
122     \_useit{\_ea\.processbooksB \printedbooks} {}
123  }
124  \_def\.processbooksA #1 {%
125     \_if\_relax#1\_relax \_else \_sxdef{pbook!#1}{}\_ea\.processbooksA \_fi
126  }
127  \_def\.processbooksB #1 {%
128     \_if\_relax#1\_relax \_else
129        \_edef\amark{#1}
130        \_edef\bmark{\_cs{f!#1}}
131        \_edef\.btit{\_cs{btit!#1}}
132        \_begingroup
133           \_edef\.currbook{#1}
134           \.newbook{#1}
135           \_wterm{** \_cs{btit!#1} {#1} **}
136           \_cs{bex!#1}
137           \_isfile{\introfile}\_iftrue \.printintro
138           \_else \.printwarn{File with introduction text \introfile\_space not found}\_fi
139  %        \.CommentedBook{#1}
140           \_isfile{\fmtfile}\_iftrue \_input{\fmtfile}
141           \_else \.printwarn{File with format info \fmtfile\_space not found}\_fi
142           \_isfile{\notesfile}\_iftrue \_input{\notesfile}
143           \_else \.printwarn{File with notes \notesfile\_space not found}\_fi
144           \_cs{bpr!#1}
145           \.bibleinput{\txsfile}
146           \_cs{bpo!#1}
```

```
147        \_endgroup
148        \_ea \.processbooksB
149     \_fi
150 }
151 \_nspublic \processbooks ;
```

`\.newbook`{⟨*a-mark*⟩} ejects previous page, prepeares header and prints the book title.

```
157 \_def\.newbook#1{\_vfil\_supereject
158    \_let\.prelinkB=\.currbook \.chapnum=0
159    \_def\.prelinkC{0}\_def\.prelinkV{0}
160    \_global\_headline={\_hfil \_ea\.setheadline\_ea{\.btit}}
161    \_line{\_hss\.bookfont\.btit\_hss}
162    \_par\_nobreak\_medskip
163 }
```

`\.setheadline`{⟨*book-title*⟩} sets `\_headline`. It is re-set for each new book by `\.newbook`.
The `\bibname` can be defined by user as a name of the translating variant of the Bible. If it is not defined then it is empty by default.

```
172 \_def\.setheadline#1{\_global\_headline={\.headfont
173    \_ifodd\_pageno
174        \_rlap{\_it\bibname\_hss}%
175        \_hfil \_the\_pageno\_hfil
176        \_hbox to\.lrmargin{\_hss\_bf#1\_ifx^\_botmark^\_else\_space \_botmark\_fi}%
177        \_kern-\.lrmargin
178    \_else
179        \_kern-\.lrmargin
180        \_hbox to\.lrmargin{\_bf#1 \_firstmark\_hss}%
181        \_hfil \_the\_pageno\_hfil
182        \_llap{\_hss\_it\bibname}%
183    \_fi
184    }
185 }
186 \_def\bibname{}
```

We want `<Fm 4>` to be a link to Fm/1:4 because it is a single-chapter book. Compare `<Gn 4>` which is a link to Gn/4:1. There is a list of single-chapter books `\nochapbooks`. User must define it. The marks of these single-chapter books are separated by spaces here. The first and the last space are added to the `\nochapbooks` macro because we need them in `\.brefBookChapter`. The `\.checknochapbooks` macro does it, moreower, it checks if the `\nochapbooks` is defined. If not, it prints warning.

```
199 \_def\.checknochapbooks {%
200    \_ifx\nochapbooks\_undefined
201        \.printwarn{\_noexpand\nochapbooks (boks without chapters) undefined.}%
202        \_def\nochapbooks{}%
203    \_else  \_edef\nochapbooks{\_space\nochapbooks\_space}\_fi
204 }
```

3 Book titles

The macro `\BookTile` ⟨*a-mark*⟩ ⟨*b-mark*⟩ {⟨*title*⟩} declares titles of each Bible books. The ⟨*a-mark*⟩ is an actual book mark used in printed text. The ⟨*b-mark*⟩ can be used in file names as `\bmark`. The mapping is done here: \def`\btit!`⟨*a-mark*⟩{⟨*title*⟩}, \def`\f!`⟨*a-mark*⟩{⟨*b-mark*⟩}.

The macro is defined as `\outer` because we don't want to see obscure errors due to missing a space after ⟨*b-mark*⟩ or ⟨*a-mark*⟩.

```
221 \_outer\_def\.BookTitle #1 #2 #3{\_sxdef{btit!#1}{#3}\_sxdef{f!#1}{#2}}
```

The `\BookException` ⟨*a-mark*⟩ {⟨*code*⟩} macro adds the ⟨*code*⟩ to the `\bex!`⟨*a-mark*⟩ macro. It is used in `\processbooks` loop in the group before files are read. You can redefine some filenames or something more special here.
Macros `\BookPre` ⟨*a-mark*⟩ {⟨*code*⟩} and `\BookPost` ⟨*a-mark*⟩ {⟨*code*⟩} are defined similarly. They add ⟨*code*⟩ to the `\bpr!`⟨*a-mark*⟩ and to the `\bpo!`⟨*a-mark*⟩ macros repectively.

```
233 \_outer\_long\_def\.BookException #1 #2{\.myaddto{bex!#1}{#2}}
234 \_outer\_long\_def\.BookPre        #1 #2{\.myaddto{bpr!#1}{#2}}
235 \_outer\_long\_def\.BookPost       #1 #2{\.myaddto{bpo!#1}{#2}}
236
237 \_nspublic \BookTitle \BookException \BookPre \BookPost ;
```

4   Actions

We create the output in two steps. First step: the data from \Note etc. are read and saved to the TEX memory. For each such data element the "action" is registered to a list of actions of the given verse. Each Bible verse has its list of actions. The second step: the Bible verses are read from a `.txs` file and all appropriate actions (registered to this verse) are processed before the verse text is printed. These actions can modify the selected parts of the verse text.

\alist!⟨*full-vref*⟩ is the list of actions associated with the verse ⟨*full-vref*⟩. The ⟨*full-vref*⟩ is full reference to the verse in the format ⟨*book-mark*⟩/⟨*chapter-num*⟩:⟨*verse-num*⟩

\.newaction{⟨*full-vref*⟩}{⟨*action-body*⟩} allocates new action.

```
257 \_def\.newaction#1#2{%
258    \_unless\_ifcsname alist!#1\_endcsname \_sdef{alist!#1}{}\_fi
259    \_ea\_addto\_csname alist!#1\_endcsname{#2}%
260 }
```

A typical "action" is \.replpre. The actions are processed for each Bible verse when the verse text is saved to the \.buff macro. The \.buff macro is processed after all actions of given verse are done.

\.replpre{⟨*prefix*⟩}{⟨*text*⟩}{⟨*fail*⟩} replaces first occurrence of ⟨*text*⟩ by ⟨*prefix*⟩{⟨*text*⟩} in \.buff macro. If the ⟨*text*⟩ is empty then ⟨*prefix*⟩{} is inserted at the beginning of the \.buff.

If ⟨*text*⟩ does not exists then ⟨*fail*⟩ is processed. It can report failed ⟨*text*⟩ by the \.text macro.

```
273 \_def\.replpre#1#2#3{%
274    \_ifx^#2^\_def\.tmp{#1{}}\_ea\_ea\_ea\_def\_ea\_ea\_ea\.buff\_ea\_ea\_ea{\_ea\.tmp\.buff}%
275    \_else
276       \_def\.replpredo##1#2##2\_end{%
277          \_ifx^##2^\_def\.text{#2}#3% <fail>
278          \_else \.replsave ##1#1{#2}##2\_end \_fi
279       }%
280       \_def\.replsave##1#2\_end{\_def\.buff{##1}}%
281       \_ea\.replpredo\.buff#2\_end
282    \_fi
283 }
```

5   The \Note macro

The first parameter of the \Note macro is ⟨*gen-vref*⟩. It is generalized reference to the Bible verse. It can be ⟨*chapter-num*⟩:⟨*verse*⟩ (the ⟨*book-mark*⟩ is appended from the \.currbook macro) or ⟨*chapter-num*⟩:⟨*verse-from*⟩-⟨*verse-to*⟩ (only ⟨*verse-from*⟩ is used for generating ⟨*gen-vref*⟩).

\.gentovref{⟨*gen-vref*⟩} expands to ⟨*full-vref*⟩.

```
297 \_def\.gentovref#1{\.currbook/\.gentovrefA#1-\end}
298 \_def\.gentovrefA#1-#2\end{#1}
```

\.renumvref ⟨*full-vref*⟩_relax does re-caclulating of ⟨*full-vref*⟩ using \renum data.

```
305 \_def\.renumvref #1/#2\_relax{#1/\_trycs{rn!\tmark!#1/#2}{#2}}
```

The ⟨*word*⟩ given as a parameter of the \Note macro (see bellow) is used as a word phrase which should be be searched in the given verse text. This parameter ⟨*word*⟩ is transformed first by expansion of \.transformword{⟨*word*⟩} to the ⟨*tword*⟩ variant and the ⟨*tword*⟩ is actually used for searching. The \.transformword{⟨*word*⟩} expands to the variant of the ⟨*word*⟩ declared by \.vdef. If not declared then it expands to the ⟨*word*⟩ itself, i.e ⟨*tword*⟩ is equal to ⟨*word*⟩ in this case.

```
316 \_def\.transformword#1{%
317    \_ifcsname v!\tmark!#1\_endcsname \_lastnamedcs
318    \_else #1\_fi
319 }
```

\Note ⟨gen-vref⟩ ⟨space⟩ {⟨word⟩} ⟨text⟩ \par transforms ⟨word⟩ to the ⟨tword⟩ (see above), saves ⟨text⟩ and activates replace-action of ⟨tword⟩ to \.doNote{⟨note-num⟩}{⟨tword⟩} in given verse.
There is an alternative syntax \Note<gen-vref> ⟨space⟩ {⟨word⟩}={⟨pword⟩} ⟨text⟩ \par If ⟨pword⟩ is given then it is printed in the note instead ⟨tword⟩. More precisely: transformed ⟨word⟩ is used for searching (and it is kept in the verse unchanged) but ⟨pword⟩ is printed in the note.
The \ww can precede \Note. If it is true then the ⟨word⟩ is prepared in \nextww and ⟨pword⟩ is in \nextwwA. Otherwise, the macros \nextww and \nextwwA are undefined.
\Note does exactly following:

- Calculates ⟨full-vref⟩ using \.gentovref{⟨genv-ref⟩} and svese it to \.fullvref.
- If the verse number of ⟨full-vref⟩ is zero, we want to insert the note-text before the chapter. This is one by the \.NoteB macro.
- Allocates new ⟨note-num⟩, i.e. \.notenum is ⟨note-num⟩.
- Modifies ⟨full-vref⟩ if \renum was declared using \.renumvref and saves the result to \.fullvrefm.
- Uses \.nextww and \.nextwwA as ⟨tword⟩ and ⟨pword⟩ if they are defined.
- Otherwise transforms ⟨word⟩ to ⟨tword⟩ by \.transformword.
- Reads ⟨pword⟩ (word to be printed in the note) by \.NoteA if the alternative syntax with ={⟨pword⟩} is used. Else ⟨pword⟩ is equal to ⟨tword⟩. Use it only if \.nextww is undefined.
- Defines \notetext!⟨note-num⟩ as ⟨text⟩.
- Defines \noteref!⟨note-num⟩ as ⟨full-vref⟩ re-caluulated by \renum.
- Defines \notepre!⟨note-num⟩ as numeric part of modified ⟨full-vref⟩. and calculates ⟨from⟩-⟨to⟩ part (if exists in ⟨gen-vref⟩) using \.renumlabel macro. This is printed prefix of the \Note.
- Defines \pword!⟨note-num⟩ as ⟨pword⟩,
- Does \.newaction{⟨full-vref⟩}{\.replpre{\.doNote{⟨note-num⟩}}
  {⟨tword⟩}{\.notefail{⟨note-num⟩}}}.
  This is done by \.AddNote{⟨full-vref⟩}{⟨note-num⟩}{⟨tword⟩}.

Note that \Note is defined as \outer in order to report correctly typical mistakes with missing empty line the text of a previous \Note.

```
365  \_newcount\.notenum
366  \_def\.Note #1 #2{%
367     \_edef\.fullvref{\.gentovref{#1}}%
368     \_ea\.isversezero\.fullvref\_iftrue
369        \_ea\.NoteB
370     \_else
371        \_incr\.notenum
372        \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
373        \_def\.tmp{#1}\.sedef{notepre!\_the\.notenum}{\_ea\.renumlabel\.fullvrefm\_relax}%
374        \_ifx\.nextww\_undefined
375           {\_def\.printwarn##1{}\_xdef\.tword{\.transformword{#2}}}%
376        \_else \_xdef\.tword{\.nextww}\_fi
377        \_afterfi{\_isnextchar={\.NoteA}{\.NoteA={}}}%
378     \_fi
379  }
380  \_def\.NoteA=#1#2% #2 separated by \par or \_par:
381
382  {%
383     \_sdef{notetext!\_the\.notenum}{\_ignorespaces#2}%
384     \.sedef{noteref!\_the\.notenum}{\.fullvrefm}%
385     \_ifx\.nextww\_undefined
386        \_ifx^#1^\_sdef{pword!\_the\.notenum\_ea}\_ea{\.tword}\_else \_sdef{pword!\_the\.notenum}{#1}\_fi
387     \_else
388        \_sdef{pword!\_the\.notenum\_ea}\_ea{\.nextwwA}%
389        \_let\.nextww=\_undefined \_let\.nextwwA=\_undefined
390     \_fi
391     \_ea\.addNote\_expanded{{\.fullvrefm}{\_the\.notenum}{\.tword}}%
392  }
393  \_def\.addNote#1#2#3{%
394     \_ifx^#3^% \.tword is empty
395        \.newaction{#1}{\_addto\.prebuff{\.doNote{#2}{}}}%
396     \_else
397        \.newaction{#1}{\.replpre{\.doNote{#2}}{#3}{\.notefail{#2}}}%
398     \_fi
399  }
```

```
400 %\_outer\_def\Note{\.Note} % will be done at the end of this macro file
```

The `\.NoteB` ⟨*text*⟩ `\par` does not register any action to the verse but defines `\chapnote!`⟨*full-vref*⟩ as the ⟨*text*⟩. This chapter note will be printed before the chapter starts.

```
409 \_def\.NoteB #1% #1 separated by \par or \_par
410
411 {%
412     \_sdef{chapnote!\.fullvref}{\_ignorespaces#1}%
413 }
414 \_def\.isversezero#1/#2:#3\_iftrue{\_ifnum #3=0 }
```

`\.renumlabel` ⟨*full-vref*⟩`\_relax` expands to the numeric part of ⟨*full-vref*⟩ and appends the `--`⟨*to*⟩ part if the `\.tmp` macro is in the format ⟨*chapter*⟩`:`⟨*from*⟩`-`⟨*to*⟩. The ⟨*to*⟩ part is re-calculated in order to the the number of verses between ⟨*from*⟩ and ⟨*to*⟩ be kept. If the ⟨*to*⟩ part is in the format ⟨*chapter*⟩`:`⟨*verse*⟩ then it is unchanged. The `\.renumlabel` macro must be expandable, so we cannot use `\isinlist` and we prepare special expandable macros `\.isdivisin` and `\.iscolonin`.

```
427 \_def\.renumlabel#1/#2\_relax{#2%
428     \_ea\.isdivisin\.tmp-\_iftrue --\_ea\.renumlabelA\.tmp\_relax#2\_relax \_fi
429 }
430 \_def\.renumlabelA#1:#2-#3\_relax#4:#5\_relax{%
431     \.iscolonin#3:\_iftrue #3\_else \_the\_numexpr#5+#3-#2\_relax \_fi
432 }
```

The `\Note` text is processed and printed in the second step, when the `.txs` file is read. Actions are assigned to each verse and they are run before the appropriate verse is printed. And `\Note` action says:

> `\.replpre{\.doNote{<note-num>}}{<tword>}{\.notefail{<note-num>}}`

It means that the ⟨*tword*⟩ is searched in the verse text and replaced by `\.doNote{`⟨*note-num*⟩`}{`⟨*tword*⟩`}`. If ⟨*tword*⟩ is not found then `\.notefail{`⟨*note-num*⟩`}` prints warning about it and `\.doNote{`⟨*note-num*⟩`}{}` is prefixed before the verse text.

```
447 \_def\.notefail#1{%
448     \.printwarn{\_csstring\\Note: \.currverse: The text "\_unexpanded\_ea{\.text}" not found}%
449     \.replpre{\.doNote{#1}}{}{}% \Note is registered with the beginning of the verse
450 }
```

The `\.doNote{`⟨*note-num*⟩`}{`⟨*tword*⟩`}` prints the real note text in the second step, when the verse text from `\.buff` is processed.

```
457 \_def\.prevtmpb{}
458 \_def\.doNote#1#2{%
459     \_edef\.tmpb{\_cs{notepre!#1}}%
460     \.notelog{\_space\_space \_csstring\\Note \.tmpb\_space {#2}={\_cs{pword!#1}} (#1)}%
461     \.noteinsert{%
462        {\_bf \_ifx\.prevtmpb\.tmpb \_else \.tmpb \_enskip \_glet\.prevtmpb=\.tmpb \_fi
463         \.trymakedest{n:\_cs{noteref!#1}}%
464         \_ea\_ifx \_csname pword!#1\_endcsname \_empty
465                   \_else \_ea\_ea\_ea\.upcasefirst \_csname pword!#1\_endcsname. \_fi}%
466        \_cs{notetext!#1}}%
467     {\notecolor#2}%
468 }
469 \_def\_printfnotemark{}
470 \_def\_textindent#1{\_noindent}
```

The phrase `{`⟨*word*⟩`}` used in notes must be exactly the same as the word used in the `.txs` text. But we want to capitalize the first letter of the ⟨*word*⟩ when printing by `\.upcasefirst`. You can say `\let\.upcasefirts=\relax` if you don't want this feature.

```
480 \_def\.upcasefirst #1{\_uppercase{#1}}
```

Because there is asynchronous processing of the `\Note` text, we have a problem when an error occurs here. We cannot reference to appropriate line where the `\Note` is written. So, we print the parameters of processed `\Note` to the log file. The user can look into this file and the last printed `\Note` parameters here refers probably to the `\Note` where the reason of the error is.

The logging is done by `\.notelog{⟨text⟩}`. It is `\wlog` by default but you can set it to `\ignoreit` or `\wterm`.

```
493 \_let\.notelog=\_wlog
```

6   Inserting data from format files

`\fmtpre` `{⟨gen-vref⟩}{⟨what⟩}` adds ⟨what⟩ to `\.fmtprebuff`, i.e. at the beginning of the verse.
`\ftmadd` `{⟨gen-vref⟩}{⟨what⟩}` adds ⟨what⟩ to `\.buff`, i.e. at the end of the verse.
`\fmtins` `{⟨gen-vref⟩}{⟨text⟩}{⟨what⟩}` inserts ⟨what⟩ after ⟨text⟩ in the verse. If ⟨text⟩ is not found then ⟨what⟩ is inserted like `\fmtpre` does it
All these commands allocate new action using `\.newaction`.

```
508 \_def\.fmtpre#1#2{\.newaction{\.gentovref{#1}}{\_addto\.fmtprebuff{#2}}}
509 \_def\.fmtadd#1#2{\.newaction{\.gentovref{#1}}{\_addto\.buff{#2}}}
510 \_def\.fmtins#1#2#3{\.newaction{\.gentovref{#1}}{\.replpre{\.fmtafter{#3}}{#2}{\.fmtfail{#3}}}}
511 \_def\.fmtafter#1#2{#2#1}
512 \_def\.fmtfail#1{\.fmtwarn\_addto\.fmtprebuff{#1}}
513 \_def\.fmtwarn{\.printwarn{\_string\fmtins: \.currverse: The text "\.text" not found}}
514
515 \_nspublic \fmtpre \fmtadd \fmtins ;
```

`\begcenter` starts the centering mode. It opens a group and does setting. User must use paired `\endcenter` in order to close this group. The `\centeringmode` status is checked by `\encenter` because curious error (about # character) should be occur without this checking.

```
524 \_newdimen\.centermargin \.centermargin=4em
525 \_def\.begcenter{\_par \_ifnum\_lastpenalty<10000 \_medskip \_fi
526     \_bgroup
527     \_def\.centeringmode{y}
528     \_parindent=0pt
529     \_leftskip=\.centermargin plus1fill
530     \_rightskip=\_leftskip
531 }
532 \_def\.endcenter{\_par
533     \_ifx\.centeringmode\_undefined
534        \.printwarn{\_noexpand\endcenter ignored: no \_noexpand\begcenter precedes}
535     \_else \_egroup \_medskip \_fi
536 }
537 \_nspublic \begcenter \endcenter ;
```

7   Printing verses from `.txs` files

When Bible text is processed then book mark is saved to `\.currbook` and each input line is separated to the ⟨chapter-num⟩:⟨verse-num⟩ and ⟨verse-text⟩.
The `\.processline` ⟨chapter⟩:⟨verse⟩⟨space⟩⟨verse-text⟩`^^J` is repeatedly processed.

```
550 \_eoldef\.processline#1{\.processverse \.currbook/#1\_end}
```

`\.processverse` ⟨full-vref⟩⟨space⟩⟨verse-text⟩`\_end` does

- defines `\.currverse` as ⟨full-vref⟩,
- prepares `\.currversenum`, `\.currversetext`, `\.currchapnum` from ⟨full-vref⟩,
- defines `\.buff` as ⟨verse-text⟩,
- processes all actions from `\alist!`⟨full-vref⟩,
- if `\.currchapnum` changed, prints new chapter by `\.printchap`
- prints verse from `\.buff` using `\.printverse`

```
564 \_newcount\.chapnum
565 \_def\.processverse #1 #2\_end{%
566    \_edef\.currverse{#1}%
567    \.preparechapverse #1
568    \_let\.prelinkV=\.currversenum
569    \_def\.buff{#2}\_def\.fmtprebuff{}\_def\.prebuff{}%
570    \_ifx\.verseto\_empty \_csname alist!#1\_endcsname \_else
571       \_fornum \.versefrom..\.verseto \_do{\_csname alist!\.currbook/\.currchapnum:##1\_endcsname}%
572    \_fi
573    \_ifnum\.currchapnum=\.chapnum \_else
574       \_let\.prelinkC=\.currchapnum \.chapnum=\.currchapnum\_relax \.printchap \_fi
575    \.printverse
576 }
577 \_def\.preparechapverse #1/#2:#3 {\_def\.currchapnum{#2}%
578    \_def\.verseto{}%
579    \.isdivisin #3-\_iftrue \.defversefromto #3\_end
580    \_else \_def\.currversenum{#3}\_let\.currversetext=\.currversenum
581    \_fi
582 }
583 \_def\.defversefromto #1-#2\_end{%
584    \_def\.versefrom{#1}\_def\.verseto{#2}%
585    \_def\.currversenum{#1}\_def\.currversetext{#1--#2}}
```

`\.printverse` prints verse from `\.currversenum` and (possibly changed) `\.buff`. It prints the single raised verse number first.

`\.printchap` prints beginning of the new chapter. `\.printbeforefirst` is a macro which is executed just before first verse of the chapter, after all material from `\fmtpre` is executed. I.e after printing a chapter name (if declared by `\fmtpre`).

```
596 \_def\.printverse{%
597    \.fmtprebuff % material accumulated by \fmtpre
598    \_ifnum\.currversenum=1 \.printbeforefirst \_fi
599    \_quitvmode \_mark{\.currchapnum:\.currversetext}%
600    \_ifx\.verseto\_empty \.trymakedest{v:\.currverse}%
601    \_else \_fornum \.versefrom..\.verseto \_do{%
602       \_wlog{xxxxx v:\.currbook/\.currchapnum:##1}\.trymakedest{v:\.currbook/\.currchapnum:##1}}%
603    \_fi
604    \_raise5pt\_hbox{\_unless\_ifnum\.currversenum=1 \.markfont\.currversetext\_fi}%
605    \.prebuff\.buff \_space
606 }
607 \_def\.printchap{\_bigskip}
608
609 \_def\.printbeforefirst{%
610    \_par\_nobreak \_medskip
611    \.printchapnote
612    \_setbox0=\_vtop{\_kern-1.5ex \_ewref\_sxdef{{ch!\.currbook/\_the\.chapnum}{\_string\.mypage}}
613                    \_hbox{\_setfontsize{at50pt}\_bf\LiRed\_the\.chapnum}}
614    \_dp0=0pt
615    \_tmpdim=\.lrmargin
616    \_advance\_tmpdim by4pt
617    \_ifnum\_the\.chapnum>9 \_advance\_tmpdim by19pt \_fi
618    \_ifodd\_trycs{ch!\.currbook/\_the\.chapnum}{0}
619       \_moveright\_tmpdim \_line{\_hss\_box0}
620    \_else \_moveleft\_tmpdim \_box0 \_fi
621    \_nobreak \_vskip-\_medskipamount
622    \_nobreak \_nointerlineskip \_noindent
623 }
624 \_def\.printchapnote{%
625    \_ifcsname chapnote!\.currbook/\_the\.chapnum:0\_endcsname
626       {\_leftskip=\_parindent plus1fill \_rightskip=\_leftskip
627        \_noindent\_it \_cs{chapnote!\.currbook/\_the\.chapnum:0}\_par}
628       \_medskip
629    \_fi
630 }
```

8 Bible references

The < will be set to active as character equivalent to the macro `\.bref`⟨*text*⟩>. This macro does all job with the hyperlinks. Fist of all, it scans the parts of the ⟨*text*⟩ and saves them to

- `\.ltextP` ... the text before a link specification (given in "...")
- `\.ltextB` ... the book mark followed by ~
- `\.ltextC` ... the chapter number followed by :
- `\.ltextV` ... the verse number
- `\.ltextS` ... sub-verse identifier (a if there is a verse 4a)
- `\.ltextF` ... the -- if the ⟨*from*⟩-⟨*to*⟩ format is given
- `\.ltextN` ... the ⟨*to*⟩ part from the ⟨*from*⟩-⟨*to*⟩ format.

All these macros above can be empty if the appropriate part of the scanned ⟨*text*⟩ is missing. The `\.linkpre` macro includes v if it is verse link, includes n if it is note link and g if it is gloss link. These macros will be converted due to `\renum` data (if needed) and printed by `\.linktext`.

op-bible.opm

```
657 \_def\.linktext{\.ltextP\.ltextB\.ltextC\.ltextV\.ltextS\.ltextF\.ltextN}
658 \_def\.bref #1>{\_let\.brefH=\_relax \_def\.linkspec{#1}\_isnextchar"{\.brefA}{\.brefA""}#1>}
659 \_def\.brefA"#1"{\_def\.ltextP{#1}%
660    \_isnextchar{ }{\_addto\.ltextP{~}\_afterassignment\.brefB\_let\.next= }%
661       {\_isnextchar{_}{\_def\.brefH{}\_afterassignment\.brefB\_let\.next= }{\.brefB}}%
662 }
663 \_def\.brefB #1>{% #1 is link-spec
664    \_def\.ltextB{}\_def\.ltextC{}\_def\.ltextF{}\_def\.ltextN{}%
665    \.isspacein #1 \_iftrue
666          \.iscolonin #1:\_iftrue \.brefBookChapterVerse #1>%
667          \_else \.brefBookChapter #1>\_fi
668    \_else \.iscolonin #1:\_iftrue \.brefChapterVerse #1>%
669    \_else \.brefVerse #1>%
670    \_fi\_fi
671    \_def\.linkpre{v}%
672    \_isnextchar n{\_def\.linkpre{n}\.brefC}%
673       {\_isnextchar g{\_def\.linkpre{g}\.brefC}%
674          {\_isnextchar a{\_def\.linkpre{a}\.brefC}%
675             {\_isnextchar i{\_def\.linkpre{i}\.brefC}{\.brefD}}}}%
676 }
677 \_def\.brefC{\_afterassignment\.brefD \_let\.next= }
678
679 \_def\.brefBookChapterVerse #1 #2:#3>{\_def\.ltextB{#1~}\.brefChapterVerse #2:#3>}
680 \_def\.brefBookChapter #1 #2>{\_def\.ltextB{#1~}%
681    \_isinlist\nochapbooks{ #1 }\_iftrue
682       \_def\.ltextC{}\_let\.ltextCin=\.ltextnCin \_afterfi{\.brefVerse #2>}%
683    \_else \_afterfi{\.brefChapter #2>}\_fi}
684 \_def\.brefChapterVerse #1:#2>{\_def\.ltextC{#1:}\.brefVerse #2>}
685 \_def\.brefVerse #1>{%
686    \.isdivisin #1-\_iftrue \.brefFromTo #1>%
687    \_else \.versedef#1\_relax\_fi
688 }
689 \_def\.brefChapter #1>{%
690    \.isdivisin #1-\_iftrue \.brefFromTo #1>\_let\.ltextC=\.ltextV
691    \_else \_def\.ltextC{#1}\_fi
692    \_def\.ltextV{}\_def\.ltextS{}%
693 }
694 \_def\.brefFromTo #1-#2>{\.versedef#1\_relax\_def\.ltextF{--}\_def\.ltextN{#2}}
```

Because the verse number can be in the format 11b, we need to separate the numeric part of this and save it to `\.ltextV` and the rest is saved to `\.ltextS`. This is done by the `\.versedef` ⟨*verse*⟩\relax macro.

op-bible.opm

```
702 \_def\.versedef {\_afterassignment\.versedefB \_tmpnum=0}
703 \_def\.versedefB #1\_relax{\_edef\.ltextV{\_the\_tmpnum}\_def\.ltextS{#1}}
```

Now, we create `\.linkfspec` from scanned data. It is ⟨*full-vref*⟩ used for hyperlinks. We must manage all situations of incomplete links.

```
710  \_def\.brefD{%
711     \_ifnum 0\.ltextV=0 \_def\.ltextV{}\_fi
712     \_if a\.linkpre \_ifx\.ltextV\_empty \_else \_edef\.ltextC{\.ltextV:}\_def\.ltextV{}\_fi\_fi
713     \_edef\.linkfspec{\_ea\.ltextBin\.ltextB~/\_ea\.ltextCin\.ltextC:/\_ea\.ltextVin\.ltextV:/}%
714     \.brefL
715  }
716  \_def\.ltextBin #1~#2/{\_ifx^#1^\.prelinkB \_else #1\_immediateassignment\_def\.prelinkB{#1}\_fi/}
717  \_def\.ltextCin #1:#2/{\_ifx^#1^\.prelinkC \_else #1\_immediateassignment\_def\.prelinkC{#1}\_fi:}
718  \_def\.ltextVin #1:#2/{\_ifx^#1^\.prelinkV \_else #1\_immediateassignment\_def\.prelinkV{#1}\_fi}
719  \_def\.ltextnCin #1:#2/{\.prelinkC:\_immediateassignment\_let\.ltextCin=\.ltextsCin}
720  \_let\.ltextsCin=\.ltextCin
```

\.prelinkB is ⟨*book-mark*⟩ of last referenced book. \.prelinkC is ⟨*chapter-num*⟩ of last referenced chapter. They are used if the reference is not full. They are initialized at the beginning of books and chapters and they are changed locally in the \Note text. If the \< is used then they are re-initialized.

```
730  \_def\<{\_let\.prelinkB=\.currbook \_let\.prelinkC=\.currchapnum \_let\.prelinkV=\.currversenum \.bref}
```

\.oncebref includes an additional macros which have to be processed in the single link, for example \reduceref. The \everybref token list includes macros which have to be applied for all links.

```
738  \_newtoks\.everybref
739  \_def\.oncebref{}
740  \_nspublic \everybref ;
```

Macro \.brefL recalculates \.linkfspec and \.linktext due to \renum data and creates the link \.linkpre:\.linkfspec with the text \.linktext.
\.renumlinktext ⟨*full-vref-ori*⟩_relax⟨*full-vref-modified*⟩_relax does re-calculation of the parts of the \.linktext macro.
The \.linkfspecone solves situation when chapter is given but no verse number: we must set the verse number to 1.
If the link destination is article, then the ⟨*full-vref*⟩ has reduced format ⟨*book*⟩/⟨*chapter*⟩. If the link destination is itroduction then the ⟨*full-vref*⟩ has more reduced format: ⟨*book*⟩/.
\.linklog {⟨*text*⟩} macro prints logging info of the link in the format

   <⟨*link-spec*⟩> = [⟨*full-vref*⟩]{⟨*printed-link*⟩}

\.linklog is \wlog by default and when \tracinglinks is set. It is \ignreit when \notracinglinks is set. You can set it to \wterm if you want.

```
761  \_def\.brefL{%
762     \_edef\.linkfspecm{\_ea\.renumvref\.linkfspec\_relax}%
763     \_ifx\.linkfspec\.linkfspecm \_else
764        \_ea\_ea\_ea\.renumlinktext \_ea\.linkfspec \_ea\_relax \.linkfspecm \_relax
765        \_let\.linkfspec=\.linkfspecm
766     \_fi
767     \_ifx\.ltextV\_empty \_ifx\.ltextC\_empty \_else \_ea\.linkfspecone \.linkfspec\_end \_fi\_fi
768     \_if a\.linkpre\_relax \_ea\.linkfspecarticle \.linkfspec\_end \_fi
769     \_if i\.linkpre\_relax \_ea\.linkfspecintro \.linkfspec\_end \_fi
770     \.linklog{\_sspace <\.linkspec>\.linkpost = [\.linkpre:\.linkfspec]%
771           {\_ifx\.brefH\_empty\_unexpanded\_ea{\.ltextP}\_else \.linktext\_fi}}%
772     \.ensuredest \.createlink
773  }
774  \_def\.linkfspecone #1:#2\_end {\_def\.linkfspec{#1:1}\_def\.prelinkV{1}}
775  \_def\.linkfspecarticle #1/#2:#3\_end {\_def\.linkfspec{#1/#2}}
776  \_def\.linkfspecintro #1/#2\_end {\_def\.linkfspec{#1/}}
777
778  \_def\.renumlinktext #1/#2:#3\_relax #4/#5:#6\_relax{%
779     \_ifx\.ltextC\_empty \_else \_def\.ltextC{#5:}\_fi
780     \_def\.ltextV{#6}%
781     \_ifx\.ltextN\_empty \_else
782        \_ifx\.ltextF\.ltextDD
783           \_isinlist\.ltextN{:}\_iftrue
784              \_ifcsname rn!\tmark!#1/\.ltextN\_endcsname \_edef\.ltextN{\_cs{rn!\tmark!#1/\.ltextN}}%
785              \_fi
786           \_else \_edef\.ltextN{\_the\_numexpr#6+\.ltextN-#3\_relax}\_fi
787        \_else \_let\.tmp=\_ignoreit % \.ltextN is a list of verses, for example 7,9,13
788           \_ea\_foreach\.ltextN,\_do ##1,{\_edef\.tmp{\.tmp,\_the\_numexpr#6+##1-#3}}%
```

```
789              \_let\.ltextN=\.tmp
790          \_fi
791      \_fi
792 }
793 \_def\.ltextDD{--}
794
795 \_def\.sspace{\_space\_space\_space\_space}
796 \_def\.linkpost{\_if v\.linkpre \_else \.linkpre\_fi \_space}
```

\tracinglinks and \notracinglinks are defined here.

```
802 \_def\tracinglinks{\_let\.linklog=\_wlog}
803 \_def\notracinglinks{\_let\.linklog=\_ignoreit}
804 \tracinglinks
```

\.createlink creates link only if it refers to the place of printed book because we don't want to see many warnings about unreferenced links when we try to print only selected books. It creates link \.linkpre:\.linkfspec with the text \.linktext

The link is created only if the book is to be printed, i.e. the \pbook!⟨book⟩ is defined.

```
815 \_def\.createlink{{%
816     \_ifx\.brefH\_empty \_let\.linktext=\.ltextP\_fi
817     \_ea\.isprintedbook\.linkfspec \_iftrue
818     \_link[\.linkpre:\.linkfspec]{\Blue}{\.linktext}%
819     \_else {\Blue\.linktext}\_fi}%
820 }
821 \_def\.isprintedbook #1/#2\_iftrue{\_ifcsname pbook!#1\_endcsname}
822 \_def\tracingouterlinks{\_def\.isprintedbook ##1\_iftrue{\_iftrue}}
```

We don't create destinations for all verses, notes etc. but only for those which are referenced. The macro \.ensuredest is called from \.createlink and it saves immediatelly \sdef{⟨link⟩:⟨full-vref⟩}{} to the special file \jobname.xrf. And the macro \pg saves immediatelly \sdef{pg:⟨link⟩:⟨full-vref⟩}{??} to this file. This .xrf file is read before standard .ref file. All link destinations save \.Xdest{⟨full-vref⟩} to the .ref file. The macro \.Xdest does nothing if \pg:⟨link⟩:⟨full-vref⟩ is not defined (from .xrf file). Otherwise, it is defined as a correct pageno. This result is used in the \pg macro. If \⟨link⟩:⟨full-vref⟩ is not defined, no link destination is crated. First TeX run creates .ref and .xrf files and does not create any hyperlink destinations. Second TeX run uses data from these files and creates correct hyperlinks and page numbers.

```
842 \_newwrite\.xrf
843 \_immediate\_openout\.xrf=\_jobname.xrf
844 \_openref
845
846 \_def\.ensuredest{\_immediate\_write\.xrf{\_string\_sdef{\.linkpre:\.linkfspec}{}}}
847 \_refdecl{
848     \_isfile{\_jobname.xrf}\_iftrue \_input{\_jobname.xrf}\_fi^^J
849     \_def\.Xdest#1{\_ifcsname pg:#1\_endcsname \_sxdef{pg:#1}{\_ea\_usesecond\_currpage}\_fi}^^J
850     \_def\.mypage{\_ea\_usesecond\_currpage}
851 }
852 \_def\.trymakedest#1{%
853     \_ifcsname #1\_endcsname \_dest[#1]\_ea\_glet\_csname #1\_endcsname \_undefined \_fi
854     \_ewref\.Xdest{{#1}}%
855 }
```

The \pg macro should be used after <...>, i.e. the \.linkpre and \.linkfspec are defined. We use them. And the page number is saved to the \pg:⟨link⟩:⟨full-vref⟩ macro in the second TeX run.

```
863 \_def\.pg{%
864     \_ifcsname pg:\.linkpre:\.linkfspec\_endcsname
865         {\_edef\.linktext{\_cs{pg:\.linkpre:\.linkfspec}}\_let\.brefH=\_relax \.createlink}%
866     \_else {\Red ??}\_fi
867     \_immediate\_write\.xrf{\_string\_sdef{pg:\.linkpre:\.linkfspec}{??}}%
868 }
869 \_nspublic \pg ;
```

9 Language variants

`\variants` ⟨*number-of-variants*⟩ {⟨*tmark-A*⟩} {⟨*tmark-B*⟩} {⟨*tmark-C*⟩} ...
sets `\.numvariants`=⟨*number-of-variants*⟩ and does `\def\tmarkA{`⟨*tmark-A*⟩`}` `\def\var!1{`⟨*tmarkA*⟩`}`
`\def\var!2{`⟨*tmark-B*⟩`}` `\def\var!3{`⟨*tmark-C*⟩`}` etc.

```
881  \_newcount\.numvariants
882  \_def\.variants{\_tmpnum=0 \_afterassignment\.variantsA \.numvariants}
883  \_def\.variantsA{%
884     \_ifnum\_tmpnum<\.numvariants
885        \_advance\_tmpnum by1
886        \_afterfi{\.variantsB{\_the\_tmpnum}}%
887     \_fi
888  }
889  \_def\.variantsB#1#2{%
890     \_ifnum#1=1 \_gdef\tmarkA{#2}\_sxdef{var!1}{#2}%
891     \_else \_sxdef{var!#1}{#2}%
892     \_fi
893     \.variantsA
894  }
895  \_nspublic \variants ;
```

`\vdef` {⟨*phrase-A*⟩} {⟨*phrase-B*⟩} {⟨*phrase-C*⟩} ... does
`\def\v!`⟨*tmark-B*⟩`!`⟨*phrase-A*⟩`{`⟨*phrase-B*⟩`}` `\def\v!`⟨*tmark-C*⟩`!`⟨*phrase-A*⟩`{`⟨*phrase-C*⟩`}` etc. Empty parameter is interpreted as undefined data. The internal macro `\.vdefB` implements the error message if there is too few parameters of `\vdef` and we were read next `\vdef`. The `\.sedef` used in the `\.vdefB`{⟨*number*⟩}{⟨*param*⟩} does real work and it defines (rougly sepaking):

```
If ⟨param⟩ is "   \def \v!⟨tmark⟩!⟨phrase-A⟩ {⟨previous param⟩}
else             \def \v!⟨tmark⟩!⟨phrase-A⟩ {⟨param⟩}
```

```
912  \_def\.vdef#1{\_def\.tmp{#1}%
913     \_ifcsname v!\_trycs{var!2}{}!\.tmp\_endcsname
914        \.printwarn{\_noexpand\vdef used secondly for phrase {\.tmp}, ignored}\_fi
915     \_tmpnum=1 \_ea\.vdefA
916  }
917  \_def\.vdefA{%
918     \_ifnum\_tmpnum<\.numvariants
919        \_advance\_tmpnum by1
920        \_afterfi{\.vdefB{\_the\_tmpnum}}%
921     \_fi
922  }
923  \_def\.vdefB#1#2{\_def\.tmpa{}%
924     \_ifx\.vdef#2\_def\.tmpa{#2}\_fi
925     \_ifx\.tmpa\_empty
926        \_ifx^#2^\_else
927           \_unless \_ifcsname v!\_cs{var!#1}!\.tmp\_endcsname
928              \.sedef{v!\_cs{var!#1}!\.tmp}{\_ifx"#2\.prevcs{#1}\.tmp \_else#2\_fi}%
929        \_fi\_fi
930        \_ea\.vdefA
931     \_else \_errmessage{\_string\vdef: too few parameters. To be read again: \_string#2}%
932        \_ea\.tmpa
933     \_fi
934  }
935  \_def\.prevcs #1#2{\_ifnum#1=2 #2\_else \_cs{v!\_cs{var!\_the\_numexpr#1-1\_relax}!#2}\_fi}
936
937  \_nspublic \vdef ;
```

`\x/`⟨*phrase*⟩`/` expands to `\v!`⟨*tmark*⟩`!`⟨*phrase*⟩ if such control sequence is defined else it expands simply to ⟨*phrase*⟩ using `\xA`. The ⟨*tmark*⟩ is actual value of the `\tmark` macro.
Note that if `\tmark` expands to ⟨*t-markA*⟩ (used in the `\variants` macro), then the `\v!`⟨*tmark*⟩`!`⟨*phrase*⟩ is not defined and the `\x` macro expands to the ⟨*phrase*⟩ directly.
`\xA` ⟨*phrase*⟩`/` expands to ⟨*phrase*⟩ and prints warning, if `\tmark` is not the first ⟨*t-markA*⟩.

```
950  \_def\.x/#1/{\_trycs{v!\tmark!#1}{\.xA#1/}}
951  \_def\.xA#1/{#1\_ifx\tmarkA\_undefined \_else \_ifx\tmark\tmarkA \_else
952     \.printwarn{\_string\x/#1/ -- this phrase is undefined by \_csstring\\vdef}%
```

```
953     \_fi\_fi
954 }
955 \_nspublic \x ;
```

\ww {⟨*phrase-A*⟩} {⟨*phrase-B*⟩} ... has the same number of parameters as \vdef. They are separated by spaces. Each parameter can be in the "single form", i.e. {⟨*phrase-A*⟩} or in the "extended form", i.e. {⟨*phrase-A*⟩}={⟨*printed-A*⟩}. The macro searchs the correct phrase (given by the \.varnum) and saves it to the \nextww. The \nextwwA is set to \nextww if there is single form of the parameter else \nextwwA is ⟨*printed-A*⟩ part of the parameter in the extended form. These macros are used in the next \Note where they are re-set to \undefined meaning.

```
968 \_def\.ww{%
969     \_ifx\.varnum\_undefined \.setvarnum \_fi
970     \_tmpnum=0
971     \_ifx\.nextww\_undefined \_ea\.wwA
972     \_else \.printwarn{Only single \_csstring\\ww must be before \_csstring\\Note}%
973         \_ea\.wwB \_fi
974 }
975 \_def\.wwA#1#2 {\_advance\_tmpnum by1
976     \_def\.nextww{#1}\_def\.nextwwA{#2}%
977     \_ifx\.nextwwA\_empty \_let\.nextwwA=\.nextww \_else \_ea \.redefwwA #2\_end \_fi
978     \_ifnum\.varnum=\_tmpnum  \_ifnum\_tmpnum<\.numvariants  \_ea\_ea\_ea \.wwB \_fi
979     \_else \_ea \.wwA \_fi
980 }
981 \_def\.wwB#1 {\_advance\_tmpnum by1
982     \_ifnum\_tmpnum<\.numvariants \_ea\.wwB \_fi
983 }
984 \_def\.redefwwA =#1\_end{\_def\.nextwwA{#1}}
985
986 % \_outer\_def\ww{\.ww} % will be done at the end of this macro file
```

The \switch macro reads a pair of parameters using \.switchA and processes the list of variants in \foreach loop. If an element from the list is equal with \tmark then the #2 (saved in \.switchD token list) is run and next parameter pairs are read by \.switchN, i.e. they are ignored.

The \Note and \ww are defined as \outer in order to better diagnose mistakes with number of parameters of \ww or missig empty line after \Note text. But we want to skip such objects in \switch parameters. This is the reason why we run \.unsetouter before the \switch parameter is read and we run \.setouter in order to return to the normal setting.

```
1001 \_newtoks\.switchD
1002 \_def\.switch {\_let\.switchN=\.switchA \.unsetouter \.switchN}
1003 \_long\_def\.switchA #1#2{\.switchD={\.setouter #2\_let\.switchN=\.switchI}%
1004     \_ifx\_relax#1\_relax \_the\.switchD
1005     \_else \_foreach #1,\_do ##1,{\_def\tmp{##1}\.switchC}%
1006     \_fi
1007     \_futurelet\.next\.switchB
1008 }
1009 \_def\.switchB{\_ifx\.next\_bgroup \.unsetouter \_ea\.switchN \_else \.setouter \_fi}
1010 \_long\_def\.switchI #1#2{\_futurelet\.next\.switchB}
1011 \_def\.switchC{\_ifx\tmp\tmark \_the\.switchD \_fi}
1012 \_def\.unsetouter{\_slet{ww}{_relax}\_slet{Note}{_relax}}
1013 \_def\.setouter{\_outer\_def\ww{\.ww}\_outer\_def\Note{\.Note}}
1014
1015 \_nspublic \switch ;
```

\.setvarnum sets the \.varnum as the position number of the current language variant due to the value of \tmark. The \variants declaration must precede.

```
1023 \_def\.setvarnum{\_gdef\.varnum{0}%
1024     \_ifnum\.numvariants=0 \_gdef\.varnum{1}\_wlog{There is only single language variant (1)}%
1025     \_else
1026         \_tmpnum=0
1027         \_loop
1028             \_advance\_tmpnum by1
1029             \_ea\_ifx \_csname var!\_the\_tmpnum\_endcsname \tmark \_xdef\.varnum{\_the\_tmpnum}\_fi
1030             \_ifnum\_tmpnum<\.numvariants \_repeat
1031         \_ifnum \.varnum=0 \_errmessage{\_noexpand\tmark isn't set, \_noexpand\.setvarnum failded}%
1032         \_else \_wlog{Language variant set by \_string\tmark{\tmark} (\.varnum)}\_fi
```

13

```
1033      \_fi
1034  }
```

\renum ⟨*book-mark*⟩ ⟨*chapter-num*⟩:⟨*verse-num*⟩ = ⟨*t-mark*⟩ ⟨*chap-num*⟩:⟨*from*⟩-⟨*to*⟩ does

```
    \def \rn!<t-mark>!<full-vref>{<chap-num>:<from>}
    \def \rn!<t-mark>!<full-vref+1>{<chap-num>:<from+1>}
    \def \rn!<t-mark>!<full-vref+2>{<chap-num>:<from+2>}
    ... etc.
    \def \rn!<t-mark>!<full-vref+n>{<chap-num>:<to>}
```

```
1048  \_def\.renum #1 #2:#3 = #4 #5:#6-#7 {%
1049      \_tmpnum=#3\_relax
1050      \_fornum #6..#7 \_do {\_sxdef{rn!#4!#1/#2:\_the\_tmpnum}{#5:##1}\_incr\_tmpnum}%
1051  }
1052  \_nspublic \renum ;
```

10   Inserting notes to the page

We declare new insert `\.noteins` used in the `\output` routine.

```
1061  \_newinsert \.noteins
1062  \_skip\.noteins=\_bigskipamount  % noterule height
1063  \_count\.noteins=500             % two columns
1064  \_dimen\.noteins=\_maxdimen      % full page of notes allowed
```

The `\.noteinsert` `{⟨text⟩}` inserts its parameter to the `\.noteins`. We open the `\insert` and set basic parameters using `\.noteset`. Then the empty box with strut height is inserted in vertical mode (in order to consecutive notes have good baselineskip between them). Then the ⟨*text*⟩ is printed and the paragraph is finalized. The empty box with strut depth is appended after the paragraph (in order to the same reason). Final `\penalty0` allows breaking between notes.

```
1077  \_def\.noteinsert #1{\_insert\.noteins{%
1078      \.noteset
1079      \_vbox to\_ht\_strutbox{}\_nobreak \_vskip-\_baselineskip
1080      #1\_unskip\_par \_nobreak \_vskip-\_baselineskip
1081      \_hbox{\_lower\_dp\_strutbox\_vbox{}}
1082      \_penalty0
1083  }}
1084  \_def\.noteset{\Heros\cond \_scalemain \_typoscale[800/800] % Heros condensed 80%
1085      \Black \_nobreak
1086      \_widowpenalty=20 \_clubpenalty=20
1087      \_leftskip=0pt \_rightskip=0pt \_parfillskip=0pt plus1fill
1088      \_parindent=0pt
1089      \_lineskiplimit=-3pt
1090      \_hsize=.5\_hsize \_advance\_hsize by-1em\_relax % two columns
1091      \_everypar{}
1092  }
```

We add macros for inserting two columns of notes from `\.noteins` into the page. First, we add `\noterule` with the space given by `\skip\.noteins`. The `\.noteins` material is prefixed by `\penalty0` (in order to allow the next `\vsplit` operation) and the `\vfil` is added (in order to the case when the second column is smaller than the first one). The `\splittopskip` is set and first `\vsplit to0pt` adds skip given by `\splittopskip` to the `\.noteins`. The `\_balancecolumns` from OpTeX for splitting to two columns is used. We need to set `\_Ncols`, `\_dimen0` and `\_box6` before running `\_balancecolumns`. We need to insert `\vskip\splittopskip` because `\_balancecolumns` supposes that the typesetting point resides at the first baseline of the columns.

The final `\vskip` does "raggedbottom". We need to add `1filll` in order to suppress the `\vfill` from the `\end` algorithm. We add `minus6pt` because the height of two columns can be by half-line higher than the insertion algorithm excepts (in the case with odd lines before splitting to the two columns).

```
1113  \_addto\_pagecontents{%
1114      \_ifvoid\.noteins \_else
1115          \_vskip\_skip\.noteins \noterule
1116          \_setbox\.noteins=\_vbox{\_penalty0 \_unvbox\.noteins \_vfil}
```

```
1117        \_splittopskip=12pt
1118        \_setbox0=\_vsplit\.noteins to0pt  % adding \splittopskip to \.noteins
1119        \_def\_Ncols{2}
1120        \_dimen0=.5\_ht\.noteins \_setbox6=\_box\.noteins
1121        \_vskip\_splittopskip
1122        \_balancecolumns
1123     \_fi
1124     \_unless\_ifvoid\.botins \_unvbox\.botins
1125     \_else \_vskip 0pt plus1filll minus8pt \_fi
1126 }
1127 \_def \noterule {\_kern-3pt {\Black \_hrule width\_hsize}\_kern 2.6pt }
```

11 Inserting images and articles to the page

`\.botins` is analogue insert as `\_topins` but the material is inserted to the bottom of the page. The material is created by `\.botinsert...\.endbot` pair of control sequences. We use it for inserting images and articles to the page.

```
1139 \_newinsert\.botins
1140 \_def\.botinsert{\_setbox0=\_vbox\_bgroup}
1141 \_def\.endbot{\_par\_egroup
1142     \_insert\.botins{\_splittopskip=0pt \_penalty100
1143        \_hrule height0pt \_nobreak\_medskip\_bigskip \_unvbox0
1144     }%
1145 }
1146 \_skip\.botins=\_zoskip     % no space added when a topinsert is present
1147 \_count\.botins=1000        % magnification factor (1 to 1)
1148 \_dimen\.botins=\_maxdimen % no limit per page
```

`\putImage` ⟨*chatper*⟩:⟨*verse*⟩ {⟨*title*⟩} [⟨*label*⟩] (⟨*params*⟩) {⟨*image-file*⟩} inserts the given image to the page where the begining of the verse given by ⟨*chapter*⟩:⟨*verse*⟩ exists. We register a new action by `\.newaction{`⟨*full-vref*⟩`}{\doImage{`⟨*title*⟩`}[`⟨*label*⟩`](`⟨*params*⟩`){`⟨*image-file*⟩`}}`. The `\doImage` puts the image by `\.botinsert...\.endbot` pair. The `\botTitle{`⟨*title*⟩`}[`⟨*label*⟩`]` prints the title of the image (or article or watever is put to the bottom of the page) and inserts the destination of hyperlink based on the ⟨*label*⟩, if the ⟨*label*⟩ isn't empty.

```
1161 \_def\.putImage #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1162     \_edef\.fullvref{\.gentovref{#1}}%
1163     \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
1164     \_ea\.newaction\_ea{\.fullvrefm}{\.doImage{#2}[#4](#6){#7}}%
1165 }
1166 \_def\.doImage #1[#2](#3)#4{% {Title}[label](params){image-file.pdf}
1167     \.botinsert
1168        \.botTitle{#1}[#2]%
1169        \_kern3pt \_nobreak
1170        \_hbox{\picw=\hsize #3\inspic{#4}}%
1171     \.endbot
1172 }
1173 \_def\.botTitle#1[#2]{\_hbox{\.captionfont
1174     \_ifx^#2^\_else \.botDest{#1}[#2]\_fi
1175     \_rlap{\Grey \_vrule height1.2em depth.5em width\_hsize}\White\_kern12pt #1}%
1176 }
1177 \_picdir={images/}
1178 \_def\.botDest#1[#2]{\_label[#2]\_wlabel{#1}}
1179
1180 \_nspublic \putImage ;
```

`\putArticle` ⟨*chapter*⟩:⟨*verse*⟩ {⟨*title*⟩} [⟨*label*⟩] (⟨*params*⟩) inserts an article given in the file `articles-*.tex` signed by `\Article` [⟨*label*⟩]. The article starts at the page where ⟨*chapter*⟩:⟨*verse*⟩ is or at the next page. The article is in two-columns style and it is divided to *k* two-columns parts each of them is inserted at the bottom of the next page.

We calculate the number of pages used for article text by following rules. All the two-columns parts have the same height. If there are more than one such a part, the height does not exceeds 2/3 of the page. But single two-column part can be higher.

`\putArticle` registers `\doArticle` using `\.newaction`. `\doArticle` is run at the beginning of given verse and creates an `\botisert`. The insert material is breakable at its beginig and between each two-column boxes created by the `\balancecolumn` macro.

We register a new action by `\.newaction{⟨full-vref⟩}{\.doArticle{⟨title⟩}[⟨label⟩](⟨params⟩)}`.

```
1204  \_newcount\.articlenum
1205  \_def\.putArticle #1 #2#3[#4]#5(#6){% chap:verse {Title} [number] (params)
1206      \_edef\.fullvref{\.gentovref{#1}}%
1207      \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
1208      \_ea\.newaction\_ea{\.fullvrefm}{\.doArticle{#2}[#4](#6)}%
1209  }
1210  \_nspublic \putArticle ;
```

The `\.doArticle {⟨Title⟩}[⟨label⟩](⟨params⟩)` inserts the article to one or more pages by `\.botinsert...\.endbot.` The Article is printed to two columns per page, all collumns of the article is completely balanced. First, the whole text is saved to the `\box0` with given column size and the number of pages is calculated in `\_tmpnum`. Then the number of columns `\_Ncols` is 2 times the number of calculated pages. The height of each two-columns part of the article is `\dimen0`. Finally we do re-boxing the output of `\_balancecolumns` in order to reach individual columns and create pairs of them by `\fornum` loop. These pairs are completed to blocks with LightGrey background. These blocks divided by `\break` are inserted into `\.botinsert`.

```
1226  \_def\.doArticle#1[#2](#3){% {Title}[number](params)
1227      \_incr\.articlenum
1228      \.botinsert
1229          \_def\.botDest##1[##2]{\.trymakedest{a:\.currbook/##2}}
1230          \_parindent=12pt \_iindent=\_parindent
1231          \_setbox0=\_vbox{\_hsize=.458\_hsize \_emergencystretch=1em
1232              \_hbadness=6000 \_baselineskip=\_dimexpr\_baselineskip plus1pt
1233              \_def\Article[##1]{\_endinput}
1234              \_penalty0
1235              \_long\_def\.searcharticle##1\Article[#2]{}
1236              \_ea\.searcharticle \_input \articlefile \_relax}
1237          \_splittopskip=12pt
1238          \_setbox1=\_vsplit0 to0pt % adding \splittopskip
1239          \_tmpdim=\_vsize \_advance\_tmpdim by-24pt % \.botTitle height plus above/below skips
1240          \_ifdim 2\_tmpdim > \_ht0 \_tmpnum=1
1241          \_else
1242              \_tmpnum=\.roundexpr{\_bp{\_ht0}/\_bp{1.333\_vsize}+0.999} % number of 2/3 pages
1243          \_fi
1244          \_multiply\_tmpnum by2 % number of columns
1245          \_edef\_Ncols{\_the\_tmpnum}
1246          \_dimen0=\_expr{1/\_Ncols}\_ht0 \_setbox6=\_box0 % height of each two-columns part
1247          \_setbox0=\_vbox{\_balancecolumns}
1248          \_tmpdim=\_ht0 \_advance\_tmpdim by1.2\_baselineskip
1249          \_setbox0=\_vbox{\_unvbox0 \_global\_setbox2=\_lastbox}
1250          \_setbox0=\_hbox{\_unhbox2
1251              \_fornum 1..\_Ncols \_do {\_unskip \_global\_setbox1##1=\_lastbox}}
1252              \_fornumstep -2: \_Ncols..1 \_do {
1253                  \_hrule height0pt\_kern5pt\_nobreak\_vfill
1254                  \_ifnum\_Ncols=##1 \.botTitle{#1}[#2]\_else \.botTitle{}[]\_fi
1255                  \_kern3pt \_nobreak
1256                  \_hbox to\_hsize{%
1257                      \_rlap{\LightGrey \_vrule height\_tmpdim depth6pt width\_hsize}%
1258                      \_kern\_parindent
1259                      \_box1##1\_hss\_box1\_the\_numexpr##1-1
1260                      \_kern\_parindent
1261                  }
1262                  \_break
1263              }
1264      \.endbot
1265  }
1266  \_def\.roundexpr#1{\_ea\_ea\_ea\.roundexprA\_expr{#1}\_relax}
1267  \_def\.roundexprA#1.#2\_relax{\_ifnum#1=0 0\_else #1\_fi}
```

12 Inserting citations to the page

`\putCite` ⟨*gen-vref*⟩ {⟨*text*⟩} creates a citation ⟨*text*⟩ inserted to the top of the page where the verse ⟨*gen-vref*⟩ is. We regiter a new action by `\.newaction{`⟨*full-vref*⟩`}{\dotopCite{`⟨*text*⟩`}}`.

```
1278  \_def\.putCite #1 #2{% chap:verse {text}
1279     \_edef\.fullvref{\.gentovref{#1}}%
1280     \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
1281     \_ea\.newaction\_ea{\.fullvrefm}{\.dotopCite{#2}}%
1282  }
1283  \_nspublic \putCite ;
```

`\.dotopCite {`⟨*text*⟩`}` creates the citation text by `\topinsert`...`\endinsert` form plain TeX. We distinguish two cases: the citation on a left page and the citation on a right page. We sawe the page position using `\_ewref` to the .ref file as `\sxdef{ct!`⟨*citenum*⟩`}{\.mypage}` and we know the page position in the second TeX run and use it in the `\ifodd` condition. The typesetting parameters differ in "left" and "right" case.

```
1295  \_newcount\.citenum
1296  \_def\.dotopCite #1{%
1297     \_topinsert
1298     \_typosize[12/16]\_bi
1299     \_incr\.citenum
1300     \_ifodd \_trycs{ct!\_the\.citenum}{0}\_relax
1301        \_leftskip=.3\_hsize plus1fil \_parfillskip=0pt
1302        \_noindent
1303        \_rlap{\_hskip\_hsize \_kern-\_leftskip \_copy\.rqqbox}\_hfill
1304     \_else
1305        \_let\quotedby=\.quotedbyright
1306        \_rightskip=.3\_hsize plus 1fil
1307        \_noindent \_llap{\_copy\.lqqbox}%
1308     \_fi
1309     {\.printCite{#1}\_unskip}\_par
1310     \_ewref\_sxdef{{ct!\_the\.citenum}{\_string\.mypage}}%
1311  %    \vskip-.3\baselineskip
1312     \_endinsert
1313  }
1314  \_def\.printCite#1{\_pdfliteral{2 Tr .15 w .9 g}#1\_pdfliteral{0 Tr 0 w 0 g}}
1315  \_def\.printCite#1{{\Grey#1}}
```

The `\.lqqbox` and `\.rqqbox` include the graphical marks for quotations. First one is used at the left pages, second one at the right pages.

The macro `\quotedby{`⟨*author*⟩`}` puts the author of the quatation to the next line. The macro `\qutedbyright` (which is used at left pages) prints the ⟨*author*⟩ at the last line if there is sufficient space.

```
1325  \_newbox\.lqqbox
1326  \_newbox\.rqqbox
1327  \_setbox\.lqqbox=\_hbox{\_lower3pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed„}}
1328  \_setbox\.rqqbox=\_hbox{\_kern2pt\_lower38pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed"}}
1329  \_ht\.lqqbox=0pt \_dp\.lqqbox=0pt
1330  \_ht\.rqqbox=0pt \_dp\.rqqbox=0pt
1331
1332  \_def\quotedby{\_par}
1333  \_def\.quotedbyright#1{%
1334     \_unskip\_nobreak\_hfill\_penalty0\_hskip2em
1335     \_null\_nobreak\_hskip\_iindent\_hbox{#1}}
```

The following macros `\Cite`, `\insertCite` and `\swapCites` are used for insertion of citations to the two-cloumn printed articles. The `\Cite`⟨*label*⟩`{`⟨*text*⟩`}` simply saves the ⟨*text*⟩ to the macro `\c!`⟨*article-num*⟩`!`⟨*label*⟩. The `\insertCite`⟨*label*⟩⟨*feft-or-right*⟩ inserts the citation declared by `\Cite` ⟨*label*⟩ to the text using `\vadjust`. The variant `\left` and `\right` is processed or ignored. This depends on the parity of the current page, which is restored from `.ref` file and saved to the macro `\cp!`⟨*article-num*⟩`!`⟨*label*⟩.

```
1349  \_def\.Cite #1#2{\_sdef{c!\_the\.articlenum!#1}{#2}}
1350  \_def\.insertCite #1#2{\_def\.citelabel{#1}%
1351     \_ifx\_left#2\.insertCiteleft
```

```
1352    \_else \_ifx#2\_right\.insertCiteright\_else
1353        \_errmessage{\_noexpand\insertCite#1: \_noexpand\left or \_noexpand\right expected}%
1354    \_fi\_fi
1355 }
1356 \_def\.insertCiteleft {%
1357    \_ifnum\.citepg=1 \.printwarn{\_noexpand\.insertCite\.citelabel: \_noexpand\.swapCites activated}\_fi
1358    \_ifodd \_numexpr\_trycs{cp!\_the\.articlenum!\.citelabel}{0}+\.citepg\_relax
1359    \_else \.insertCitelr \_left \_fi
1360 }
1361 \_def\.insertCiteright{%
1362    \_ifodd \_numexpr\_trycs{cp!\_the\.articlenum!\.citelabel}{0}+\.citepg\_relax
1363    \.insertCitelr \_right \_fi
1364 }
1365 \_def\.insertCitelr#1{\_unskip\_vadjust{\_vbox{%
1366    \_ewref\_sxdef{{cp!\_the\.articlenum!\.citelabel}{\_string\.mypage}}%
1367    \_vskip6pt
1368    \_advance\_hsize by\_parindent
1369    \_typosize[12/16]\_bi\Grey
1370        \_ifx#1\_left
1371            \_def\quotedby{\_par\_hfill}
1372            \_rightskip=\_parindent plus1fil \_leftskip=0pt
1373            \_setbox0\_vbox{%
1374                \_medskip \_noindent
1375                \_llap{\_copy\.lqqbox}\_ignorespaces
1376                \.printCite{\_cs{c!\_the\.articlenum!\.citelabel}}\_medskip}%
1377            \_hbox{\_kern-\_parindent\_rlap{\White
1378                \_vrule height\_ht0 width\_hsize}\_box0}%
1379        \_else
1380            \_leftskip=\_parindent plus1fil
1381            \_parfillskip=0pt
1382            \_setbox0\_vbox{%
1383                \_medskip \_noindent
1384                \_rlap{\_hskip\_hsize\_kern-\_parindent\_copy\.rqqbox}\_hfill
1385                \_ignorespaces \.printCite{\_cs{c!\_the\.articlenum!\.citelabel}}\_medskip}%
1386            \_rlap{\_rlap{\White \_vrule height\_ht0 width\_hsize}\_box0}%
1387        \_fi
1388    \_vskip6pt
1389 }}}
1390 \_def\.swapCites{\_def\.citepg{1}}
1391 \_def\.citepg{0}
1392
1393 \_nspublic \Cite \insertCite ;
```

Insertions into the intro text

```
1401 %% TBN page 236
1402
1403 \_newcount\.shapenum
1404 \_newdimen\.ii \_newdimen\.w
1405 \_def\.oblom #1 od #2 odsadit #3 {\_par \.ii=#1 \.w=\_hsize
1406    \_ifdim\.ii>\_zo \_advance\.w by-\.ii
1407    \_else \_advance\.w by\.ii \.ii=\_zo \_fi
1408    \.shapenum=1 \_tmpnum=0 \_def\.shapelist{}
1409    \_loop \_ifnum\.shapenum<#2 \_edef\.shapelist{\.shapelist\_zo\_hsize}%
1410        \_advance\.shapenum by1 \_repeat
1411    \_loop \_edef\.shapelist{\.shapelist\.ii\.w}%
1412        \_advance\_tmpnum by1 \_ifnum\_tmpnum<#3 \_repeat
1413    \_advance\.shapenum by#3 \_edef\.shapelist{\.shapelist\_zo\_hsize}
1414    \.doshape}
1415 \_def\.doshape{\_parshape \.shapenum \.shapelist}
1416 \_newcount\.globpar
1417 \_ifx\_partokenset \_undefined \_def\.partoken{\par} \_else \_def\.partoken{\_par} \_fi
1418 \_def\.doshape{\_global\.globpar=0 \_ea\_def\.partoken{\_ifhmode\.shapepar\_fi}}
1419 \_def\.shapepar{\_prevgraf=\.globpar \_parshape\.shapenum\.shapelist
1420    \_endgraf \_global\.globpar=\_prevgraf
1421    \_ifnum \_prevgraf>\.shapenum \_ea\_let\.partoken=\_endgraf \_fi
1422 }
1423
1424 \_def\.Citehereleft #1 (#2) #3{{
1425    \_par
```

18

```
1426        \_def\quotedby{\_par\_hfill}
1427        \_rightskip=\_parindent plus1fil \_leftskip=0pt
1428        \_setbox0\_vbox{{%
1429            \_typosize[12/16]\_bi\Grey
1430            \_hsize=.5\_hsize
1431            \_medskip \_noindent
1432            \_llap{\_copy\.lqqbox}\_ignorespaces
1433            \.printCite{#3}\_medskip}}%
1434    \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1435    \_xdef\.lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1436    \_nointerlineskip\_vbox to0pt{\_kern#1\_baselineskip #2
1437        \_hbox{\_rlap{\White
1438            \_kern-3mm\_vrule height\_ht0 width.5\_hsize}\_box0}%
1439    \_vss}}
1440    \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1441    \.oblom {.5\_tmpdim} od #1 odsadit {\.lines}
1442 }
1443 \_def\.Citehereright #1 (#2) #3{{
1444    \_par
1445        \_def\quotedby{\_par\_parfillskip=0pt \_hfill}
1446        \_leftskip=\_parindent plus1fill \_rightskip=0pt
1447        \_setbox0\_vbox{{%
1448            \_typosize[12/16]\_bi\Grey
1449            \_hsize=.5\_hsize
1450            \_vskip\_medskipamount \_rlap{\_kern\_hsize\_copy\.rqqbox}\_vskip-\_medskipamount
1451            \.printCite{\_noindent\_ignorespaces#3}\_medskip}}%
1452    \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1453    \_xdef\.lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1454    \_nointerlineskip\_vbox to0pt{\_kern#1\_baselineskip #2
1455        \_hbox to\_hsize{\_hss
1456            \_llap{\White \_vrule height\_ht0 width.5\_hsize \_kern-3mm}%
1457            \_llap{\_box0}}
1458    \_vss}}
1459    \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1460    \.oblom {-.5\_tmpdim} od #1 odsadit {\.lines}
1461 }
1462
1463 \_def\.Citehere{\_par \_ifodd\_pageno \_ea\.Citehereright \_else \_ea\.Citehereleft \_fi}
1464
1465 \_nspublic \Citehere ;
1466
1467 \_def\.insertBot #1#2[#3]#4(#5)#6{% {Title} [label] (params) {data}
1468    \.botinsert
1469        \.botTitle{#1}[#3]%
1470        \_kern3pt \_nobreak
1471        \_vbox{\_picwidth=\_hsize #5 #6}%
1472    \.endbot
1473 }
1474 \_def\.putBot #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1475    \_edef\.fullvref{\.gentovref{#1}}%
1476    \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
1477    \_ea\.newaction\_ea{\.fullvrefm}{\.insertBot{#2}[#4](#6){#7}}%
1478 }
1479
1480 \_def\.c[#1/#2]#3{% text podel krivky: \c[init-rotace/repetice]{text}
1481    \_pdfsave\_pdfrotate{#1}\_rlap{\_edef\.tmpb{#3}\_replstring\.tmpb{ }{{ }}\_def\.tmpa{#2}%
1482                \_ea\_foreach\.tmpb\_do{##1\.tmpa}}\_pdfrestore \_kern10mm
1483 }
1484 \_nspublic \insertBot \putBot \c ;
```

\.printintro macro (by default) prints the itroduction of th book from the \introfile, prints the title
"Introduction" (depending on the current language and puts all introduction text between \.begblock
and \.endblock.

```
1493 \_def\.printintro{%
1494    \.begblock
1495        \_dest[i:\.currbook/]
1496        \.chaptit{\_mtext{intro}}%
1497        \_input{\introfile}
```

```
1498        \.endblock
1499 }
```

Text block with grey background splittable to more pages is between `\.begblock` and `\.endblock` macros. It is used for introduction text. See also OpTeX trick 0031.

```
1507 \_newcount\.blocklevel  % nesting level of blocks
1508 \_def\.begblock{\_par\_bgroup
1509     \_advance\.blocklevel by1 \_advance\_leftskip by\_iindent \_rightskip=\_leftskip
1510     \_medskip
1511     \_pdfsavepos \_ea\_wref\_ea\.Xblock\_ea{\_ea{\_the\.blocklevel}B{\_the\_pdflastypos}}
1512     \_nobreak \_medskip
1513 }
1514 \_def\.endblock{\_par\_nobreak\_medskip
1515     \_pdfsavepos \_ea\_wref\_ea\.Xblock\_ea{\_ea{\_the\.blocklevel}E{\_the\_pdflastypos}}
1516     \_medskip \_egroup
1517 }
1518 \_refdecl{%
1519     \_def\.Xblock#1#2#3{\_ifnum#1=1 \_edef\.tmp{frm:\_ea\_ignoresecond\_currpage}^^J
1520         \_unless\_ifcsname \.tmp \_endcsname \_sxdef{\.tmp}{}\_fi^^J
1521         \_sxdef{\.tmp}{\_cs{\.tmp}#2{#3}}\_fi}
1522 }
1523 \_newdimen\.frtop \_newdimen\.frbottom % positions of top and bottom text on the pages
1524 \_def\.frcolor{.93 g } % light grey -- color of blocks.
1525 \_pgbackground={%
1526     \_slet{_opb_tmp}{frm:\_the\_gpageno}
1527     \_ifx\.tmp\_undefined \_def\.tmp{}\_fi
1528     \.frtop=\_dimexpr \_pdfpageheight-\_voffset+\_smallskipamount\_relax
1529     \.frbottom=\_dimexpr\_pdfpageheight-\_voffset-\_vsize-\_medskipamount\_relax
1530     \_ifx\.frnext y \_edef\.tmp{B{\_number\.frtop}\.tmp}\_global\_let\.frnext n\_fi
1531     \_ea\.printframes \.tmp B{0}E{\_number\.frbottom}
1532     \_ifx\.frameslist\_empty \_else
1533     \_pdfliteral{q \.frcolor 1 0 0 1 0 \_bp{-\_pdfpageheight} cm \.frameslist Q}\_fi
1534 }
1535 \_def\.printframes B#1#2E#3{\_ifnum#1=0 \_else
1536     \.printframe {\_hoffset}{#3sp}{\_xhsize}{\_ifnum#1=-1 \_number\.frtop\_else#1\_fi sp-#3sp}
1537     \_ifx^#2^\_else \_global\_let\.frnext=y \_let\.printframes=\_relax \_fi
1538     \_ea\.printframes\_fi
1539 }
1540 \_def\.frameslist{}
1541 \_def\.printframe #1#2#3#4{\_edef\.frameslist{\.frameslist
1542     \_bp{#1} \_bp{#2} \_bp{#3} \_bp{#4} re f }%
1543 }
```

13   Outline

```
1551 \_newdimen\.colsep
1552 \.colsep=10pt
1553
1554 \_def\.Outline{
1555     \_medskip
1556 %   \_filbreak
1557     \.chaptit{\_mtext{outline}}%
1558     \_everylist={\_ifcase\_ilevel \_or \_style I \_or \_style A \_or \_style n \_fi}
1559     \_sdef{_item:A}{\_strut\_uppercase\_ea{\_athe\_itemnum}. }
1560     \_sdef{_item:I}{\_strut\_uppercase\_ea{\_romannumeral\_itemnum}. }
1561     \_hsize=.5\_hsize \_advance\_hsize by-\.colsep
1562     \_emergencystretch=40pt
1563     \_leftskip=0pt \_rightskip=0pt
1564 }
1565 \_def\.rightnote#1{\_par
1566     \_setbox0=\_hbox{\_kern\_hsize \_kern\.colsep
1567                     \_vtop{\_leftskip=0pt \_kern0pt\_noindent\_strut\_it#1}}
1568     \_ht0=0pt \_dp0=0pt \_box0 \_nointerlineskip
1569 }
1570 \_nspublic \Outline \rightnote ;
```

14 Typesetting variants

By default, chapter numbers are in the outer margin and quotes characters too. The `\normalchapnums` macro moves chater numbers to the left side in the first paragraph, cquotes characters are removed and outer margins are reduced because there is no material in them.

```
1584  \_def\.normalchapnumbers{
1585     \_margins/2 a2 (25,25,20,20)mm
1586     \.lrmargin=0pt
1587     \_setbox0=\_box\.lqqbox \_setbox0=\_box\.rqqbox
1588     \_def\.printbeforefirst{%
1589        \_nobreak\_medskip
1590        \.printchapnote
1591        \_hangindent=\_parindent \_hangafter=-2
1592        \_noindent \_llap{\_vbox to0pt
1593           {\_kern-8pt\_hbox{\_setfontsize{at23pt}\_bf\Red\_the\.chapnum\_kern5pt}\_vss}}%
1594     }
1595  }
1596  \_nspublic \normalchapnumbers ;
```

15 Checking syntax

```
1604  \_def\.checksyntax#1 {%
1605     \_let\processbooks=\_relax
1606     \_ifx\_relax#1\_relax \_else
1607        \_begingroup
1608           \_the\.syntaxmacros
1609           \_wterm{^^J** checking file: #1 **^^J}
1610           \_input{#1}
1611           \_vfil\_break
1612        \_endgroup
1613     \_ea\.checksyntax \_fi
1614  }
1615
1616  \_newtoks\.syntaxmacros
1617  {\_catcode`<=13
1618  \_global\.syntaxmacros={
1619  \_def<#1>{\_bgroup
1620     \_message{checking \_unexpanded{<#1>}}%
1621     \_ifx\_relax#1\_relax \_errmessage{empty link}\.nobref\_else \_afterfi{\.checkbref#1>\.bref#1>}\_fi
1622     \_glet\.linkpre=\.linkpre \_glet\.linkfspec=\.linkfspec
1623     \_egroup
1624  }
1625  \_def\.checkbref#1#2>{%
1626     \_isinlist{.#1#2}{<}\_iftrue \_errmessage{duplicated \_string<}\.nobref\_else
1627     \_ifx"#1\.checkbrefQ #1#2>\_else \.checkbrefD #1#2>\_fi\_fi
1628  }
1629  \_def\.checkbrefQ "#1"#2#3>{\.checkbrefD #2#3>}
1630  \_def\.checkbrefD #1>{%
1631     \_isinlist{.#1}{ }\_iftrue\.checkbrefS#1>\_else\.checkbrefN#1>\_fi
1632  }
1633  \_def\.checkbrefS #1 #2>{\.checkbrefN#2>}
1634  \_def\.checkbrefN #1>{%
1635     \_def\.tmpb{#1}
1636     \_ifx\.tmpb\_empty \_errmessage{missing link data}\.nobref\_else
1637        \_replstring\.tmpb{:}{}\_replstring\.tmpb{-}{}\_replstring\.tmpb{_}{}%
1638        \_replstring\.tmpb{a}{}\_replstring\.tmpb{b}{}\_replstring\.tmpb{c}{}%
1639        \_setbox0=\_hbox{\_tmpnum=0\.tmpb\_relax}%
1640        \_ifdim\_wd0>0pt \_errmessage{nonnumeric link data}\.nobref\_fi
1641     \_fi
1642  }
1643  \_def\.nobref{\_def\.bref##1>{{\Red\_string<##1>}}}
1644  \_def\.currbook{}
1645  \_def\.prelinkB{BK}
1646  \_def\.prelinkC{BK}
1647  \_def\.prelinkV{0}
1648  \_def\nochapbooks{BK}
```

```
1649  \_let\<=<
1650
1651  \_def\x/#1/{\_def\.tmpb{#1}%
1652     \_isinlist\.tmpb\x\_iftrue \.badx
1653     \_else \_isinlist\.tmp<\_iftrue \.badx
1654     \_else \_isinlist\.tmp\enditems\_iftrue \.badx \_else \.x/#1/\_fi\_fi\_fi
1655  }
1656  \_def\.badx{\_errmessage{unclosed \_string\x/.../}}
1657
1658  \_def\Article[#1]{}
1659  \_def\Cite #1 {\_par\_noindent{\_bf Cite: }}
1660  \_def\insertCite #1#2{}
1661
1662  \_def\putArticle #1 #2[#3]#4(#5){}
1663  \_def\putCite #1:#2 {\_par\_noindent{\_bf Cite: }}
1664  \_def\putBot #1 #2[#3]#4(#5){\_vbox}
1665
1666  \_def\c[#1/#2]#3{#3}
1667
1668  \_long\_ea\_def\_csname Note\_endcsname #1 #2#3%
1669
1670     {\_par \_let\.nextww\_undefined \_noindent{\_bf Note #1:} #3\_par}
1671  }}
1672  \_nspublic \checksyntax ;
```

16   TODO macros

The temporary macros are here. I plan to rewrite them.

```
1682  \_def\.chaptit#1{\_line{\_hss\.chapfont\Red#1\_hss}
1683     \_nobreak
1684  }
1685  \_def\schaptit#1{\_bigskip\.chaptit{#1}\_medskip}
1686
1687  \_nspublic \chaptit ;
1688
1689  \_sdef{_mt:intro:en}{Introduction}    \_sdef{_mt:outline:en}{Outline}
1690  \_sdef{_mt:intro:cs}{Úvod}            \_sdef{_mt:outline:cs}{Osnova}
1691
1692  \_def\dopsat{{\Red !!! DOPSAT !!! }}
1693
1694  \_def\.bibleinput#1 {\_bgroup
1695     \_catcode`##=13 \_bgroup\_lccode`~=`## \_lowercase{\_egroup\_let~}=\.processline
1696     \_input{#1}%
1697     \_egroup
1698  }
1699  \_let\FormatedBook=\_ignoreit  % for backward compatibility
1700  \_let\CommentedBook=\_ignoreit % for backward compatibility
```

Active character < used for references.

```
1706  \_outer\_def\Note {\.Note}
1707  \_outer\_def\ww   {\.ww}
1708
1709  \_def\_afterload{\_adef<{\.bref}}
1710  \_afterload
1711
1712  \_endnamespace
```

17   Index

22