

# OP-Bible – Technical Documentation

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

`op-bible.opm`

```
3 \_codedec1 \processbooks {OpBible: macros for creating annotated Bible}
```

## 1 Preparatory work

Loading packages.

`op-bible.opm`

```
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.

`op-bible.opm`

```
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 {\_setcmkcolor{0 0 0 .1}}
36 \_def\LiRed {\_setcmkcolor{0 .2 .2 0}}
```

Fonts.

`op-bible.opm`

```
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`  $\{\langle text \rangle\}$  prints warning. `\.sedef`  $\{\langle name \rangle\}\{\langle body \rangle\}$  is expanded `\sdef`. `\.myaddto`  $\{\langle macro-name \rangle\}\{\langle text \rangle\}$  adds  $\langle text \rangle$  to  $\langle macro-name \rangle$  globally. Moreover it defines the undefined macro by `\sdef{\langle macro-name \rangle}\{\langle text \rangle\}`.

`op-bible.opm`

```
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}
```

## 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.
- 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 `\BookExceptions` 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 T<sub>E</sub>X memory.
- Inputs notes file if it exists. The notes are saved to the T<sub>E</sub>X 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 T<sub>E</sub>X memory.
- Calls `\bpo!<a-mark>` in order to apply `\BookPost` data.

Note that the macros `\introfile`, `\fmtfile`, and `\notesfile` give the location of appropriate 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 T<sub>E</sub>X memory is freed.

op-bible.opm

```

107 \_def\.processbooks {\_par
108   \.checknochapbooks
109   \_useit{\_ea\.processbooksA \printedbooks} {}
110   \_useit{\_ea\.processbooksB \printedbooks} {}
111 }
112 \_def\.processbooksA #1 {%
113   \_if\_relax#1\_relax \_else \_sxddef{pbook!#1}{\_ea\.processbooksA \_fi
114 }
115 \_def\.processbooksB #1 {%
116   \_if\_relax#1\_relax \_else
117     \_edef\amark{#1}
118     \_edef\bmark{\_cs{f!#1}}
119     \_edef\.btit{\_cs{btit!#1}}
120     \_begingroup
121       \.newbook{#1}
122       \_wterm{** \_cs{btit!#1} {#1} **}
123       \_cs{bex!#1}
124       \_isfile{\introfile}\_iftrue \.printintro
125       \_else \.printwarn{File with introduction text \introfile\_space not found}\_fi
126       \_CommentedBook{#1}
127       \_isfile{\fmtfile}\_iftrue \_input{\fmtfile}
128       \_else \.printwarn{File with format info \fmtfile\_space not found}\_fi
129       \_isfile{\notesfile}\_iftrue \_input{\notesfile}
130       \_else \.printwarn{File with notes \notesfile\_space not found}\_fi
131       \_cs{bpr!#1}
132       \.bibleinput{txsfile}
133       \_cs{bpo!#1}
134     \_endgroup
135     \_ea \.processbooksB
136   \_fi
137 }
138 \_nspublic \processbooks ;

```

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

op-bible.opm

```

144 \_def\.newbook#1{\_vfif\_supereject
145   \_edef\.currbook{#1}\_let\_prelinkB=\.currbook \.chapnum=0
146   \_def\.prelinkC{0}\_def\.prelinkV{0}

```

```

147 \_global\_headline={\_hfil \_ea\setheadline\_ea{\_btit}}
148 \_line{\_hss\bookfont\_btit\_hss}
149 \_par\_nobreak\_medskip
150 }

```

`\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.

```

159 \_def\setheadline#1{\_global\_headline={\_headfont
160 \_ifodd\_pageno
161 \_rlap{\_it\bibname\_hss}%
162 \_hfil \_the\_pageno\_hfil
163 \_hbox to\lmargin{\_hss\_bf#1\_ifx\_botmark\_else\_space \_botmark\_fi}%
164 \_kern-\lmargin
165 \_else
166 \_kern-\lmargin
167 \_hbox to\lmargin{\_bf#1 \_firstmark\_hss}%
168 \_hfil \_the\_pageno\_hfil
169 \_llap{\_hss\_it\bibname}%
170 \_fi
171 }
172 }
173 \_def\bibname{}

```

op-bible.opm

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, moreover, it checks if the `\nochapbooks` is defined. If not, it prints warning.

```

186 \_def\checknochapbooks {%
187 \_ifx\nochapbooks\_undefined
188 \_printwarn{\_noexpand\nochapbooks (boks without chapters) undefined.}%
189 \_def\nochapbooks{}%
190 \_else \_edef\nochapbooks{\_space\nochapbooks\_space}\_fi
191 }

```

op-bible.opm

### 3 Book titles

The macro `\BookTitle <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\bf!<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>`.

```

208 \_outer\_def\BookTitle #1 #2 #3{\_sxdef{btit!#1}{#3}\_sxdef{bf!#1}{#2}}

```

op-bible.opm

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 respectively.

```

220 \_outer\_long\_def\BookException #1 #2{\_myaddto{bex!#1}{#2}}
221 \_outer\_long\_def\BookPre #1 #2{\_myaddto{bpr!#1}{#2}}
222 \_outer\_long\_def\BookPost #1 #2{\_myaddto{bpo!#1}{#2}}
223
224 \_nspublic \BookTitle \BookException \BookPre \BookPost ;

```

op-bible.opm

## 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!` $\langle full-vref \rangle$  is the list of actions associated with the verse  $\langle full-vref \rangle$ . The  $\langle full-vref \rangle$  is full reference to the verse in the format  $\langle book-mark \rangle / \langle chapter-num \rangle : \langle verse-num \rangle$   
`\.newaction!` $\langle full-vref \rangle$  $\{ \langle action-body \rangle \}$  allocates new action.

op-bible.opm

```
244 \_def\.newaction#1#2{%
245   \_unless\_ifcurname alist!#1\_endcurname \_sdef{alist!#1}{\_fi
246   \_ea\_addto\_csname alist!#1\_endcurname{#2}%
247 }
```

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!` $\langle prefix \rangle$  $\{ \langle text \rangle \}$  $\{ \langle fail \rangle \}$  replaces first occurrence of  $\langle text \rangle$  by  $\langle prefix \rangle \{ \langle text \rangle \}$  in `\.buff` macro. If the  $\langle text \rangle$  is empty then  $\langle prefix \rangle \{ \}$  is inserted at the beginning of the `\.buff`.

If  $\langle text \rangle$  does not exists then  $\langle fail \rangle$  is processed. The  $\langle fail \rangle$  macro can use `\.text` where  $\langle text \rangle$  is saved.

op-bible.opm

```
260 \_def\.replpre#1#2#3{%
261   \_ifx^#2\_def\.tmp{#1}{\_ea\_ea\_ea\_def\_ea\_ea\_ea\.buff\_ea\_ea\_ea{\_ea\.tmp\.buff}%
262   \_else
263     \_def\.replpredo##1#2##2\_end{%
264       \_ifx^##2\_def\.text{#2}#3% <fail>
265       \_else \.replsave ##1#1{#2}##2\_end \_fi
266     }%
267     \_def\.replsave##1#2\_end{\_def\.buff{##1}}%
268     \_ea\.replpredo\.buff#2\_end
269   \_fi
270 }
```

## 5 The \Note macro

The first parameter of the `\Note` macro is  $\langle gen-vref \rangle$ . It is generalized reference to the Bible verse. It can be  $\langle chapter-num \rangle : \langle verse \rangle$  (the  $\langle book-mark \rangle$  is appended from `\CommentedBook` token list) or  $\langle chapter-num \rangle : \langle verse-from \rangle - \langle verse-to \rangle$  (only  $\langle verse-from \rangle$  is used for generating  $\langle gen-vref \rangle$ ).

`\.gentovref!` $\langle gen-vref \rangle$  expands to  $\langle full-vref \rangle$ .

op-bible.opm

```
284 \_newtoks\CommentedBook
285 \_def\.gentovref#1{\the\CommentedBook/\.gentovrefA#1-\end}
286 \_def\.gentovrefA#1-#2\end{#1}
287
288 \_nspublic \CommentedBook ;
```

`\.renumvref`  $\langle full-vref \rangle$ `\_relax` does re-calculating of  $\langle full-vref \rangle$  using `\renum` data.

op-bible.opm

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

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

op-bible.opm

```
305 \_def\.transformword#1{%
306   \_ifcurname v!\tmark!#1\_endcurname \_lastnamedcs
307   \_else #1\_fi
308 }
```

`\Note`  $\langle gen-vref \rangle$   $\langle space \rangle$   $\{ \langle word \rangle \}$   $\langle text \rangle$  `\par` transforms  $\langle word \rangle$  to the  $\langle tword \rangle$  (see above), saves  $\langle text \rangle$  and activates replace-action of  $\langle tword \rangle$  to `\.doNote!` $\langle note-num \rangle$  $\{ \langle tword \rangle \}$  in given verse.

There is an alternative syntax `\Note`  $\langle gen-vref \rangle$   $\langle space \rangle$   $\{ \langle word \rangle \} = \{ \langle pword \rangle \}$   $\langle text \rangle$  `\par` If  $\langle pword \rangle$  is

given then is is printed in the note instead  $\langle tword \rangle$ . More precisely: transformed  $\langle word \rangle$  is used for searching (and it is kept in the verse unchanged) but  $\langle pword \rangle$  is printed in the note.

The  $\backslash ww$  can precede  $\backslash Note$ . If it is true then the  $\langle word \rangle$  is prepared in  $\backslash nextww$  and  $\langle pword \rangle$  is in  $\backslash nextwwA$ . Otherwise, the macros  $\backslash nextww$  and  $\backslash nextwwA$  are undefined.

$\backslash Note$  does exactly following:

- Allocates new  $\langle note-num \rangle$ ,
- Transforms  $\langle gen-vref \rangle$  to  $\langle full-vref \rangle$  using  $\backslash .gentovref$ .
- Modifies  $\langle full-vref \rangle$  if  $\backslash renum$  was declared using  $\backslash .renumvref$  and saves the result to  $\backslash .fullvrefm$ .
- Use  $\backslash nextww$  and  $\backslash nextwwA$  as  $\langle tword \rangle$  and  $\langle pword \rangle$  if they are defined.
- Otherwise transforms  $\langle word \rangle$  to  $\langle tword \rangle$  by  $\backslash .transformword$ .
- Reads  $\langle pword \rangle$  (word to be printed in the note) if the alternative syntax with  $=\{\langle pword \rangle\}$  is used. Else  $\langle pword \rangle$  is equal to  $\langle tword \rangle$ .
- Defines  $\backslash notetext!\langle note-num \rangle$  as  $\langle text \rangle$ .
- Defines  $\backslash noteref!\langle note-num \rangle$  as  $\langle full-vref \rangle$ .
- Defines  $\backslash notepre!\langle note-num \rangle$  as numeric part of modified  $\langle full-vref \rangle$  and calculates  $\langle from \rangle$ - $\langle to \rangle$  part (if exists in  $\langle gen-vref \rangle$ ) using  $\backslash renumlabel$  macro. This is printed prefix of the  $\backslash Note$ .
- Defines  $\backslash pword!\langle note-num \rangle$  as  $\langle pword \rangle$ ,
- Does  
 $\backslash .newaction\{\langle full-vref \rangle\}\{\backslash .replpre\{\backslash .doNote\{\langle note-num \rangle\}\}\{\langle tword \rangle\}\{\backslash .notefail\{\langle note-num \rangle\}\}\}.$

op-bible.opm

```

346 \_newcount\_.notenumber
347 \_outer\_def\_.Note #1 #2{%
348   \_edef\_.fullvref{\_.gentovref{#1}}%
349   \_ea\_.isversezero\_.fullvref\_iftrue
350   \_ea\_.NoteB
351   \_else
352     \_incr\_.notenumber
353     \_edef\_.fullvrefm{\_.ea\_.renumvref\_.fullvref\_relax}%
354     \_def\_.tmp{#1}\_.sedef{notepre!\_.the\_.notenumber}\_.ea\_.renumlabel\_.fullvrefm\_relax}%
355     \_ifx\_.nextww\_undefined
356       {\_def\_.printwarn##1{\_xdef\_.tword{\_.transformword{#2}}}%
357       \_else \_xdef\_.tword{\_.nextww}\_fi
358       \_afterfi{\_isnextchar={\_.NoteA}\_.NoteA={}}}%
359   \_fi
360 }
361 \_def\_.NoteA=#1#2% #2 separated by \par or \_par:
362
363 {%
364   \_sdef{notetext!\_.the\_.notenumber}\_.ignorespaces#2}%
365   \_.sedef{noteref!\_.the\_.notenumber}\_.fullvrefm}%
366   \_ifx\_.nextww\_undefined
367     \_ifx^#1^\_sdef{pword!\_.the\_.notenumber\_ea\_ea\_.tword}\_else \_sdef{pword!\_.the\_.notenumber}\_.fi
368   \_else
369     \_sdef{pword!\_.the\_.notenumber\_ea\_ea\_.nextwwA}%
370     \_let\_.nextww=\_undefined \_let\_.nextwwA=\_undefined
371   \_fi
372   \_ea\_.addNote\_expanded{\_.fullvrefm}\_.the\_.notenumber\_.tword}%
373 }
374 \_def\_.addNote#1#2#3{%
375   \_ifx^#3^ \_.tword is empty
376   \_.newaction{#1}\_.addto\_.prebuff{\_.doNote{#2}\_}%
377   \_else
378   \_.newaction{#1}\_.replpre{\_.doNote{#2}\_}\_.notefail{#2}\_}%
379   \_fi
380 }
381 \_def\_.NoteB #1% #1 separated by \par or \_par
382
383 {%
384   \_sdef{!chapnote:\_.fullvref}\_.ignorespaces#1}%
385 }
386 \_def\_.isversezero#1/#2:#3\_iftrue{\_ifnum #3=0 }
387
388 \_nspublic \_Note ;

```

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

op-bible.opm

```
401 \_def\renumlabel#1/#2\_relax#2%
402   \_ea\isdivisin\tmp-\_iftrue --\_ea\renumlabelA\tmp\_relax#2\_relax \_fi
403 }
404 \_def\renumlabelA#1:#2-#3\_relax#4:#5\_relax{%
405   \iscolonin#3:\_iftrue #3\_else \_the\_numexpr#5+#3-#2\_relax \_fi
406 }
```

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  $\langle tword \rangle$  is searched in the verse text and replaced by `\.doNote{<note-num>}{<tword>}`. If  $\langle tword \rangle$  is not found then `\.notefail{<note-num>}` prints warning about it and `\.doNote{<note-num>}{}` is prefixed before the verse text.

op-bible.opm

```
421 \_def\.notefail#1{%
422   \.printwarn{\_csstring\\Note: \.currverse: The text "\_unexpanded\_ea{\.text}" not found}%
423   \.replpre{\.doNote{#1}}{}}{\_Note is registered with the beginning of the verse
424 }
```

And the `\.doNote{<note-num>}{<tword>}` prints the real note text in the second step, when the verse text from `\.buff` is processed.

op-bible.opm

```
431 \_def\prevtmpb{}
432 \_def\.doNote#1#2{%
433   \_edef\tmpb{\_cs{notepre!#1}}%
434   \.notelog{\_space\_space \_csstring\\Note \.tmpb\_space {#2}={\_cs{pword!#1}} {#1}}%
435   \.noteinsert{%
436     {\_bf \_ifx\prevtmpb\tmpb \_else \.tmpb \_enskip \_glet\prevtmpb=\tmpb \_fi
437     \.trymakedest{n:\_cs{noteref!#1}}%
438     \_ea\_ifx \_csname pword!#1\_endcsname \_empty
439     \_else \_ea\_ea\_ea\upcasefirst \_csname pword!#1\_endcsname. \_fi}%
440     \_cs{notetext!#1}}%
441     {\notecolor#2}}%
442 }
443 \_def\_printfnotemark{}
444 \_def\_textindent#1{\_noindent}
```

The phrase  $\langle word \rangle$  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  $\langle word \rangle$  when printing. You can say `\let\upcasefirsts=\relax` if you don't want this feature.

op-bible.opm

```
453 \_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`.

op-bible.opm

```
466 \_let\.notelog=\_wlog
```

## 6 Inserting data from format files

`\fmtpre {<gen-vref>}{<what>}` adds  $\langle what \rangle$  to `\.fmtprebuff`, i.e. at the beginning of the verse.

`\ftmadd {<gen-vref>}{<what>}` adds  $\langle what \rangle$  to `\.buff`, i.e. at the end of the verse.

`\fmtins {<gen-vref>}{<text>}{<what>}` inserts  $\langle what \rangle$  after  $\langle text \rangle$  in the verse. If  $\langle text \rangle$  is not found then

$\langle what \rangle$  is inserted like `\fmtpre` does it  
All these commands allocate new action using `\.newaction`.

op-bible.opm

```

481 \_let\FormatedBook=\.CommentedBook
482 \_def\.fmtpre#1#2{\.newaction{\.gentovref{#1}}{\_addto\.fmtprebuff{#2}}}
483 \_def\.fmtadd#1#2{\.newaction{\.gentovref{#1}}{\_addto\.buff{#2}}}
484 \_def\.fmtins#1#2#3{\.newaction{\.gentovref{#1}}{\.replpre{\.fmtafter{#3}}{#2}{\.fmtfail{#3}}}}
485 \_def\.fmtafter#1#2{#2#1}
486 \_def\.fmtfail#1{\.fmtwarn\_addto\.fmtprebuff{#1}}
487 \_def\.fmtwarn{\.printwarn{\_stringfmtins: \.currverse: The text "\_unexpanded\_ea{\.text}" not found}}
488
489 \_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.

op-bible.opm

```

498 \_newdimen\centermargin \centermargin=4em
499 \_def\.begcenter{\_par \_ifnum\_lastpenalty<10000 \_medskip \_fi
500 \_bgroup
501 \_def\centeringmode{y}
502 \_parindent=0pt
503 \_leftskip=\centermargin plus1fill
504 \_rightskip=\leftskip
505 }
506 \_def\.endcenter{\_par
507 \_ifx\centeringmode\_undefined
508 \_printwarn{\_noexpand\endcenter ignored: no \_noexpand\begcenter precedes}
509 \_else \_egroup \_medskip \_fi
510 }
511 \_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  $\langle chapter-num \rangle$ : $\langle verse-num \rangle$  and  $\langle verse-text \rangle$ .

The `\processverse`  $\langle full-vref \rangle$  $\langle space \rangle$  $\langle verse-text \rangle$ `\_end` is repeatedly processed.

op-bible.opm

```

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

```

`\.processverse`  $\langle full-vref \rangle$  $\langle space \rangle$  $\langle verse-text \rangle$ `\_end` does

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

op-bible.opm

```

538 \_newcount\chapnum
539 \_def\processverse #1 #2\_end{%
540 \_edef\currverse{#1}%
541 \_preparechapverse #1
542 \_let\prelinkV=\currversenum
543 \_def\buff{#2}\_def\.fmtprebuff{\_def\prebuff{}}%
544 \_ifx\verseto\_empty \_csname alist!#1\_endcsname \_else
545 \_for num \.versefrom..\verseto \_do{\_csname alist!\.currbook/\.currchapnum:#1\_endcsname}%
546 \_fi
547 \_ifnum\currchapnum=\chapnum \_else
548 \_let\prelinkC=\currchapnum \_chapnum=\currchapnum\_relax \.printchap \_fi
549 \.printverse
550 }
551 \_def\preparechapverse #1/#2:#3 {\_def\currchapnum{#2}%
552 \_def\verseto{}}%
553 \.isdivisin #3-\_iftrue \.defversefromto #3\_end
554 \_else \_def\currversenum{#3}\_let\currversetext=\currversenum

```



```

555 \_fi
556 }
557 \_def\versefromto #1-#2\_end{%
558 \_def\versefrom{#1}\_def\verseto{#2}%
559 \_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`).

op-bible.opm

```

570 \_def\.printverse{%
571 \_fmtprebuff % material accumulated by \fmtpre
572 \_ifnum\.currversenum=1 \.printbeforefirst \_fi
573 \_quitvmode \_mark{\.currchapnum:\.currversetext}%
574 \_ifx\verseto\_empty \.trymakedest{v:\.currverse}%
575 \_else \_for num \versefrom..\verseto \_do{%
576 \_wlog{xxxxx v:\.currbook/\.currchapnum:##1}\.trymakedest{v:\.currbook/\.currchapnum:##1}}%
577 \_fi
578 \_raise5pt\_hbox{\_unless\_ifnum\.currversenum=1 \.markfont\.currversetext\_fi}%
579 \.prebuff\.buff \_space
580 }
581 \_def\.printchap{\_bigskip}
582
583 \_def\.printbeforefirst{%
584 \_par\_nobreak \_medskip
585 \.printchapnote
586 \_setbox0=\_vtop{\_kern-1.5ex \_ewref\_sxdef{ch!\.currbook/\_the\.chapnum}{\_string\.mypage}}
587 \_hbox{\_setfontsize{at50pt}\_bf\LiRed\_the\.chapnum}}
588 \_dp0=0pt
589 \_tmpdim=\_lrmargin
590 \_advance\_tmpdim by4pt
591 \_ifnum\_the\.chapnum>9 \_advance\_tmpdim by19pt \_fi
592 \_ifodd\_trycs{ch!\.currbook/\_the\.chapnum}{0}
593 \_moveright\_tmpdim \_line{\_hss\_box0}
594 \_else \_moveleft\_tmpdim \_box0 \_fi
595 \_nobreak \_vskip-\_medskipamount
596 \_nobreak \_nointerlineskip \_noindent
597 }
598 \_def\.printchapnote{%
599 \_ifcsname !chapnote:\.currbook/\_the\.chapnum:0\_endcsname
600 {\_leftskip=\_parindent plus1fill \_rightskip=\_leftskip
601 \_noindent\_it \_cs{!chapnote:\.currbook/\_the\.chapnum:0}\_par}
602 \_medskip
603 \_fi
604 }

```

## 8 Bible references

We prepare temporary macros first.

`\.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.

op-bible.opm

```

616 \_def\.isspacein #1 #2\_iftrue{\_isempty{#2}\_iffalse}
617 \_def\.iscolonin #1:#2\_iftrue{\_isempty{#2}\_iffalse}
618 \_def\.isdivisin #1-#2\_iftrue{\_isempty{#2}\_iffalse}

```

The `<` will be set to active as character equivalent to the macro `\.bref<text>`. This macro does all job with the hyperlinks. First 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  $\langle from \rangle$ - $\langle to \rangle$  format is given
- `\.ltextN` ... the  $\langle to \rangle$  part from the  $\langle from \rangle$ - $\langle to \rangle$  format.

All these macros above can be empty if the appropriate part of the scanned  $\langle text \rangle$  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

```

641 \_def\.linktext{\.ltextP\.ltextB\.ltextC\.ltextV\.ltextS\.ltextF\.ltextN}
642 \_def\.bref #1>{\_let\.brefH=\_relax \_def\.linkspec{#1}\_isnextchar{"\.brefA}{\.brefA"}#1>}
643 \_def\.brefA"#1"{\_def\.ltextP{#1}%
644   \_isnextchar{ }{\_addto\.ltextP{~}\_afterassignment\.brefB\_let\.next= }%
645   {\_isnextchar_{}\_def\.brefH{\\_afterassignment\.brefB\_let\.next= }{\.brefB}}%
646 }
647 \_def\.brefB #1>{% #1 is link-spec
648   \_def\.ltextB{\\_def\.ltextC{\\_def\.ltextF{\\_def\.ltextN{\}
649     \_isspacein #1 \_iftrue
650       \_iscolonin #1:\_iftrue \.brefBookChapterVerse #1>%
651       \_else \.brefBookChapter #1>\_fi
652     \_else \_iscolonin #1:\_iftrue \.brefChapterVerse #1>%
653     \_else \.brefVerse #1>%
654     \_fi\_fi
655     \_def\.linkpre{v}%
656     \_isnextchar n{\_def\.linkpre{n}\.brefC}%
657     {\_isnextchar g{\_def\.linkpre{g}\.brefC}%
658     {\_isnextchar a{\_def\.linkpre{a}\.brefC}%
659     {\_isnextchar i{\_def\.linkpre{i}\.brefC}{\.brefD}}}%
660 }
661 \_def\.brefC{\_afterassignment\.brefD \_let\.next= }
662
663 \_def\.brefBookChapterVerse #1 #2:#3>{\_def\.ltextB{#1~}\.brefChapterVerse #2:#3>}
664 \_def\.brefBookChapter #1 #2>{\_def\.ltextB{#1~}%
665   \_isinlist\nochapbooks{ #1 } \_iftrue
666     \_def\.ltextC{\_let\.ltextCin=\.ltextnCin \_afterfi{\.brefVerse #2>}%
667     \_else \_afterfi{\.brefChapter #2>}\_fi}
668 \_def\.brefChapterVerse #1:#2>{\_def\.ltextC{#1:}\.brefVerse #2>}
669 \_def\.brefVerse #1>{%
670   \_isdivisin #1-\_iftrue \.brefFromTo #1>%
671   \_else \.versedef#1\_relax\_fi
672 }
673 \_def\.brefChapter #1>{%
674   \_isdivisin #1-\_iftrue \.brefFromTo #1>\_let\.ltextC=\.ltextV
675   \_else \_def\.ltextC{#1}\_fi
676 \_def\.ltextV{\\_def\.ltextS{\}
677 }
678 \_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`  $\langle verse \rangle$ `\relax` macro.

op-bible.opm

```

686 \_def\.versedef {\_afterassignment\.versedef \_tmpnum=0}
687 \_def\.versedefB #1\_relax{\_edef\.ltextV{\_the\_tmpnum}\_def\.ltextS{#1}}

```

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

op-bible.opm

```

694 \_def\.brefD{%
695   \_ifnum 0\.ltextV=0 \_def\.ltextV{\\_fi
696   \_if a\.linkpre \_ifx\.ltextV\_empty \_else \_edef\.ltextC{\.ltextV{\\_def\.ltextV{\\_fi\_fi
697   \_edef\.linkfspec{\_ea\.ltextBin\.ltextB~/\_ea\.ltextCin\.ltextC:/\_ea\.ltextVin\.ltextV:/}%
698   \.brefL
699 }
700 \_def\.ltextBin #1-#2/{\_ifx^#1^\.prelinkB \_else #1\_immediateassignment\_def\.prelinkB{#1}\_fi/}
701 \_def\.ltextCin #1:#2/{\_ifx^#1^\.prelinkC \_else #1\_immediateassignment\_def\.prelinkC{#1}\_fi:}
702 \_def\.ltextVin #1:#2/{\_ifx^#1^\.prelinkV \_else #1\_immediateassignment\_def\.prelinkV{#1}\_fi}
703 \_def\.ltextnCin #1:#2/{\_prelinkC:\_immediateassignment\_let\.ltextCin=\.ltextS}
704 \_let\.ltextS=\.ltextCin

```

`\.prelinkB` is  $\langle book-mark \rangle$  of last referenced book. `\.prelinkC` is  $\langle chapter-num \rangle$  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.

```
714 \_def<{\_let\.prelinkB=\.currbook \_let\.prelinkC=\.currchapnum \_let\.prelinkV=\.currversenum \.brief}
```

`\.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.

```
722 \_newtoks\everybref
723 \_def\.oncebref{}
724 \_nspublic \everybref ;
```

Macro `\.brefL` recalculates `\.linkfspec` and `\.linktext` due to `\renum` data and creates the link `\.linkpre:\.linkfspec` with the text `\.linktext`.

`\.renumlinktext  $\langle full-vref-ori \rangle$  \_relax  $\langle full-vref-modified \rangle$  \_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  $\langle full-vref \rangle$  has reduced format  $\langle book \rangle / \langle chapter \rangle$ . If the link destination is introduction then the  $\langle full-vref \rangle$  has more reduced format:  $\langle book \rangle /$ .

`\.linklog  $\{ \langle text \rangle \}$`  macro prints logging info of the link in the format

$\langle \langle link-spec \rangle \rangle = [ \langle full-vref \rangle ] \{ \langle printed-link \rangle \}$

`\.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.

```
745 \_def\.brefL{%
746   \_edef\.linkfspecm{\_ea\.renumvref\.linkfspec\_relax}%
747   \_ifx\.linkfspec\.linkfspecm \_else
748     \_ea\_ea\_ea\.renumlinktext \_ea\.linkfspec \_ea\_relax \.linkfspecm \_relax
749     \_let\.linkfspec=\.linkfspecm
750   \_fi
751   \_ifx\ltextV\_empty \_ifx\ltextC\_empty \_else \_ea\.linkfspecone \.linkfspec\_end \_fi\_fi
752   \_if a\.linkpre\_relax \_ea\.linkfspecarticle \.linkfspec\_end \_fi
753   \_if i\.linkpre\_relax \_ea\.linkfspecintro \.linkfspec\_end \_fi
754   \.linklog{\sspace <\_unexpanded\_ea{\.linkspec}>\.linkpost = [\linkpre:\linkspec]%
755     {\_ifx\ltextH\_empty\_unexpanded\_ea{\ltextP}\_else\_unexpanded\_ea{\.linktext}\_fi}}%
756   \.ensuredest \.createlink
757 }
758 \_def\.linkfspecone #1/#2\_end {\_def\.linkfspec{#1:1}\_def\.prelinkV{1}}
759 \_def\.linkfspecarticle #1/#2:#3\_end {\_def\.linkfspec{#1/#2}}
760 \_def\.linkfspecintro #1/#2\_end {\_def\.linkfspec{#1/}}
761
762 \_def\.renumlinktext #1/#2:#3\_relax #4/#5:#6\_relax{%
763   \_ifx\ltextC\_empty \_else \_def\ltextC{#5}\_fi
764   \_def\ltextV{#6}%
765   \_ifx\ltextN\_empty \_else
766     \_ifx\ltextF\ltextDD
767       \_isinlist\ltextN{:}\_iftrue
768         \_ifcsname rn!\tmark!#1/\ltextN\_endcsname \_edef\ltextN{\_cs{rn!\tmark!#1/\ltextN}}\_fi
769       \_else \_edef\ltextN{\_the\_numexpr#6+.\ltextN-#3\_relax}\_fi
770     \_else \_let\ltextN=\_ignoreit % \ltextN is a list of verses, for example 7,9,13
771     \_ea\_foreach\ltextN,\_do #1,{\_edef\ltextN{\_the\_numexpr#6+##1-#3}}%
772     \_let\ltextN=\ltextN
773   \_fi
774   \_fi
775 }
776 \_def\ltextDD{--}
777
778 \_def\sspace{\_space\_space\_space\_space}
779 \_def\linkpost{\_if v\.linkpre \_else \.linkpre\_fi \_space}
```

`\tracinglinks` and `\notracinglinks` are defined here.

```
785 \_def\tracinglinks{\_let\linklog=\wlog}
786 \_def\notracinglinks{\_let\linklog=\ignoreit}
787 \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. The link is created always if a user declared `\tracingallrefs`.

op-bible.opm

```
799 \_def\_.createlink{%
800   \_ifx\_.briefH\_empty \_let\_.linktext=\.ltextP\_fi
801   \_ea\_.isprintedbook\_.linkfspec \_iftrue
802   \_link[\.linkpre:\.linkfspec]{\Blue}{\_.linktext}%
803   \_else {\Blue\_.linktext}\_fi}%
804 }
805 \_def\_.isprintedbook #1/#2\_iftrue{\_ifcsname pbook!#1\_endcsname}
806 \_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 immediately `\sdef{⟨link⟩:⟨full-vref⟩}{}` to the special file `\jobname.xrf`. And the macro `\pg` saves immediately `\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 `.rfx` 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 created. First `TeX` run creates `.ref` and `.xrf` files and does not create any hyperlink destinations. Second `TeX` run uses data from these files and create correct hyperlinks and page numbers.

op-bible.opm

```
826 \_newwrite\_.xrf
827 \_immediate\_openout\_.xrf=\_jobname.xrf
828 \_openref
829
830 \_def\_.ensuredest{\_immediate\_write\_.xrf{\_string\_sdef{\.linkpre:\.linkfspec}{}}}
831 \_refdecl{
832   \_isfile{\_jobname.xrf}\_iftrue \_input{\_jobname.xrf}\_fi^^J
833   \_def\_.Xdest#1{\_ifcsname pg:#1\_endcsname \_sxdef{pg:#1}{\_ea\_usesecond\_currrpage}\_fi}^^J
834   \_def\_.mypage{\_ea\_usesecond\_currrpage}
835 }
836 \_def\_.trymakedest#1{%
837   \_ifcsname #1\_endcsname \_dest[#1]\_ea\_glet\_csname #1\_endcsname \_undefined \_fi
838   \_ewref\_.Xdest{#1}}%
839 }
```

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.

op-bible.opm

```
847 \_def\_.pg{%
848   \_ifcsname pg:\.linkpre:\.linkfspec\_endcsname
849   {\_edef\_.linktext{\_cs{pg:\.linkpre:\.linkfspec}}\_let\_.briefH=\_relax \.createlink}%
850   \_else {\Red ??}\_fi
851   \_immediate\_write\_.xrf{\_string\_sdef{pg:\.linkpre:\.linkfspec}{??}}%
852 }
853 \_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.

op-bible.opm

```
865 \_newcount\_.numvariants
866 \_def\_.variants{\_tmpnum=0 \_afterassignment\_.variantsA \_.numvariants}
867 \_def\_.variantsA{%
868   \_ifnum\_tmpnum<\_.numvariants
869     \_advance\_tmpnum by1
870     \_afterfi{\_.variantsB{\_the\_tmpnum}}%
871   \_fi
872 }
```

```

873 \_def\.\variantsB#1#2{%
874   \_ifnum#1=1 \_gdef\tmarkA{#2}\_sxdef{var!1}{#2}%
875   \_else \_sxdef{var!#1}{#2}%
876   \_fi
877   \.variantsA
878 }
879 \_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 (roughly sepaking):

```

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

```

op-bible.opm

```

896 \_def\.\vdef#1{\_def\.\tmp{#1}%
897   \_ifcsname v!\_trycs{var!2}{!}\.\tmp\_endcsname
898   \.printwarn{\_noexpand\vdef used secondly for phrase {\.\tmp}, ignored}\_fi
899   \_tmpnum=1 \_ea\.\vdefA
900 }
901 \_def\.\vdefA{%
902   \_ifnum\_tmpnum<\.numvariants
903     \_advance\_tmpnum by1
904     \_afterfi{\.\vdefB{\_the\_tmpnum}}%
905   \_fi
906 }
907 \_def\.\vdefB#1#2{\_def\.\tmpa{#1}%
908   \_ifx\.\vdef#2\_def\.\tmpa{#2}\_fi
909   \_ifx\.\tmpa\_empty
910     \_ifx^#2^\_else
911       \_unless \_ifcsname v!\_cs{var!#1}!\.\tmp\_endcsname
912         \.sedef{v!\_cs{var!#1}!\.\tmp}{\_ifx"#2\.\prevcs{#1}\.\tmp \_else#2\_fi}%
913       \_fi\_fi
914       \_ea\.\vdefA
915     \_else \_errmessage{\_string\vdef: too few parameters. To be read again: \_string#2}%
916     \_ea\.\tmpa
917   \_fi
918 }
919 \_def\.\prevcs #1#2{\_ifnum#1=2 #2\_else \_cs{v!\_cs{var!\_the\_numexpr#1-1\_relax}!#2}\_fi}
920
921 \_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⟩`.

op-bible.opm

```

934 \_def\.\x/#1/{\_trycs{v!\tmark!#1}{\xA#1/}}
935 \_def\.\xA#1/{#1\_ifx\tmarkA\_undefined \_else \_ifx\tmark\tmarkA \_else
936   \.printwarn{\_string\x/#1/ -- this phrase is undefined by \_csstring\vdef}%
937   \_fi\_fi
938 }
939 \_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 searches 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.

op-bible.opm

```

952 \_outer\_def\.\ww{%
953   \_ifx\.\varnum\_undefined \.setvarnum \_fi
954   \_tmpnum=0

```

```

955 \_ifx\.\nextww\_undefined\_ea\.\wwA
956 \_else \.printwarn{Only single \_csstring\ww must be before \_csstring\Note}%
957 \_ea\.\wwB\_fi
958 }
959 \_def\.\wwA#1#2 {\_advance\_tmpnum by1
960 \_def\.\nextww{#1}\_def\.\nextwwA{#2}%
961 \_ifx\.\nextwwA\_empty\_let\.\nextwwA=\.\nextww\_else\_ea\_redefwwA #2\_end\_fi
962 \_ifnum\.\varnum=\_tmpnum\_ifnum\_tmpnum<\.numvariants\_ea\_ea\_ea\_wwB\_fi
963 \_else\_ea\_wwA\_fi
964 }
965 \_def\.\wwB#1 {\_advance\_tmpnum by1
966 \_ifnum\_tmpnum<\.numvariants\_ea\.\wwB\_fi
967 }
968 \_def\.\redefwwA=#1\_end{\_def\.\nextwwA{#1}}
969
970 \_nspublic \ww ;

```

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.

op-bible.opm

```

985 \_newtoks\.\switchD
986 \_def\.\switch {\_let\.\switchN=\.\switchA\_unsetouter\_switchN}
987 \_long\_def\.\switchA #1#2{\_switchD={\_setouter #2\_let\.\switchN=\.\switchI}%
988 \_ifx\_relax#1\_relax\_the\.\switchD
989 \_else\_foreach #1,\_do ##1,{\_def\tmp{##1}\.\switchC}%
990 \_fi
991 \_futurelet\.\next\.\switchB
992 }
993 \_def\.\switchB{\_ifx\.\next\_bgroup\_unsetouter\_ea\.\switchN\_else\_setouter\_fi}
994 \_long\_def\.\switchI #1#2{\_futurelet\.\next\.\switchB}
995 \_def\.\switchC{\_ifx\tmp\tmark\_the\.\switchD\_fi}
996 \_def\.\unsetouter{\_slet{ww}{\_relax}\_slet{Note}{\_relax}}
997 \_def\.\setouter{\_slet{ww}{\_opb\_iww}\_slet{Note}{\_opb\_iNote}}
998 \_let\.\iww=\.\ww % backup of outer ww
999 \_let\.\iNote=\.\Note % backup of outer Note
1000
1001 \_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.

op-bible.opm

```

1009 \_def\.\setvarnum{\_gdef\.\varnum{0}%
1010 \_ifnum\.\numvariants=0\_gdef\.\varnum{1}\_wlog{There is only single language variant (1)}%
1011 \_else
1012 \_tmpnum=0
1013 \_loop
1014 \_advance\_tmpnum by1
1015 \_ea\_ifx\_csname var!\_the\_tmpnum\_endcsname \tmark\_xdef\.\varnum{\_the\_tmpnum}\_fi
1016 \_ifnum\_tmpnum<\.numvariants\_repeat
1017 \_ifnum\.\varnum=0\_errmessage{\_noexpand\tmark isn't set, \_noexpand\.\setvarnum failed}%
1018 \_else\_wlog{Language variant set by \_string\tmark\tmark (\.\varnum)}\_fi
1019 \_fi
1020 }

```

`\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>}

```

```

1034 \_def\renum #1 #2:#3 = #4 #5:#6-#7 {%
1035   \_tmpnum=#3\_relax
1036   \_forum #6..#7 \_do {\_sxdef{rn!#4!#1/#2:\_the\_tmpnum}{#5:#1}\_incr\_tmpnum}%
1037 }
1038 \_nspublic \renum ;

```

## 10 Inserting notes to the page

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

op-bible.opm

```

1047 \_newinsert \.noteins
1048 \_skip\.noteins=\_bigskipamount % noterule height
1049 \_count\.noteins=500 % two columns
1050 \_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.

op-bible.opm

```

1063 \_def\.noteinsert #1{\_insert\.noteins{%
1064   \.noteset
1065   \_vbox to\_ht\_strutbox{\\_nobreak \_vskip-\_baselineskip
1066   #1\_unskip\_par \_nobreak \_vskip-\_baselineskip
1067   \_hbox{\_lower\_dp\_strutbox\_vbox{}}
1068   \_penalty0
1069 }}
1070 \_def\.noteset{\Heros\cond \_scalemain \_typoscale[800/800] % Heros condensed 80%
1071   \Black \_nobreak
1072   \_widowpenalty=20 \_clubpenalty=20
1073   \_leftskip=0pt \_rightskip=0pt \_parfillskip=0pt plus1fill
1074   \_parindent=0pt
1075   \_lineskiplimit=-3pt
1076   \_hsize=.5\_hsize \_advance\_hsize by-1em\_relax % two columns
1077   \_everypar{}
1078 }

```

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 expects (in the case with odd lines before splitting to the two columns).

op-bible.opm

```

1099 \_addto\_pagecontents{%
1100   \_ifvoid\.noteins \_else
1101     \_vskip\_skip\.noteins \noterule
1102     \_setbox\.noteins=\_vbox{\_penalty0 \_unvbox\.noteins \_vfil}
1103     \_splittopskip=12pt
1104     \_setbox0=\_vsplit\.noteins to0pt % adding \splittopskip to \.noteins
1105     \_def\_Ncols{2}
1106     \_dimen0=.5\_ht\.noteins \_setbox6=\_box\.noteins
1107     \_vskip\_splittopskip
1108     \_balancecolumns
1109     \_fi
1110     \_unless\_ifvoid\.botins \_unvbox\.botins
1111     \_else \_vskip 0pt plus1filll minus8pt \_fi
1112   }
1113 \_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.

op-bible.opm

```
1125 \_newinsert\.botins
1126 \_def\.botinsert{\_setbox0=\_vbox\_bgroup}
1127 \_def\.endbot{\_par\_egroup
1128   \_insert\.botins{\_splittopskip=0pt \_penalty100
1129     \_hrule height0pt \_nobreak\_medskip\_bigskip \_unvbox0
1130   }%
1131 }
1132 \_skip\.botins=\_zoskip    % no space added when a topinsert is present
1133 \_count\.botins=1000      % magnification factor (1 to 1)
1134 \_dimen\.botins=\_maxdimen % no limit per page
```

`\putImage <chapter>:<verse> {<title>} [<label>] (<params>) {<image-file>}` inserts the given image to the page where the beginning 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 whatever is put to the bottom of the page) and inserts the destination of hyperlink based on the `<label>`, if the `<label>` isn't empty.

op-bible.opm

```
1147 \_def\.putImage #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1148   \_edef\.fullvref{\_gentovref{#1}}%
1149   \_edef\.fullvrefm{\_ea\_renumvref\.fullvref\_relax}%
1150   \_ea\.newaction\_ea{\_fullvrefm}{\doImage{#2}[#4] (#6){#7}}%
1151 }
1152 \_def\.doImage #1[#2] (#3)#4{% {Title}[label] (params){image-file.pdf}
1153   \_botinsert
1154     \_botTitle{#1}[#2]%
1155     \_kern3pt \_nobreak
1156     \_hbox{\picw=\hsize #3\inspic{#4}}%
1157   \_endbot
1158 }
1159 \_def\.botTitle#1[#2]{\_hbox{\_captionfont
1160   \_ifx^#2\_else \_botDest{#1}[#2]\_fi
1161   \_rlap{\Grey \_vrule height1.2em depth.5em width\_hsize}\White\_kern12pt #1}%
1162 }
1163 \_picdir={images/}
1164 \_def\.botDest#1[#2]{\_label[#2]\_wlabel{#1}}
1165
1166 \_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>)}`.

op-bible.opm

```
1190 \_newcount\.articlenum
1191 \_def\.putArticle #1 #2#3[#4]#5(#6){% chap:verse {Title} [number] (params)
1192   \_edef\.fullvref{\_gentovref{#1}}%
1193   \_edef\.fullvrefm{\_ea\_renumvref\.fullvref\_relax}%
1194   \_ea\.newaction\_ea{\_fullvrefm}{\doArticle{#2}[#4] (#6)}%
1195 }
1196 \_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 `\for` loop. These pairs are completed to blocks with LightGrey background. These blocks divided by `\break` are inserted into `\.botinsert`.

op-bible.opm

```

1212 \_def\doArticle#1[#2](#3){% {Title}[number] (params)
1213   \_incr\articlenum
1214   \.botinsert
1215   \_def\botDest##1[##2]{\trymakedest{a:\currbook/##2}}
1216   \_parindent=12pt \_iindent=\_parindent
1217   \_setbox0=\_vbox{\_hsize=.458\_hsize \_emergencystretch=1em
1218     \_hbadness=6000 \_baselineskip=\_dimexpr\_baselineskip plus1pt
1219     \_def\Article[##1]{\_endinput}
1220     \_penalty0
1221     \_long\_def\searcharticle##1\Article[#2]}
1222     \_ea\searcharticle \_input \articlefile \_relax}
1223   \_splittopskip=12pt
1224   \_setbox1=\_vsplit0 to0pt % adding \splittopskip
1225   \_tmpdim=\_vsize \_advance\_tmpdim by-24pt % \.botTitle height plus above/below skips
1226   \_ifdim 2\_tmpdim > \_ht0 \_tmpnum=1
1227   \_else
1228     \_tmpnum=\_roundexpr{\_bp{\_ht0}/\_bp{1.333\_vsize}+0.999} % number of 2/3 pages
1229   \_fi
1230   \_multiply\_tmpnum by2 % number of columns
1231   \_edef\_Ncols{\_the\_tmpnum}
1232   \_dimen0=\_expr{1/\_Ncols}\_ht0 \_setbox6=\_box0 % height of each two-columns part
1233   \_setbox0=\_vbox{\_balancecolumns}
1234   \_tmpdim=\_ht0 \_advance\_tmpdim by1.2\_baselineskip
1235   \_setbox0=\_vbox{\_unvbox0 \_global\_setbox2=\_lastbox}
1236   \_setbox0=\_hbox{\_unhbox2
1237     \_for num 1..\_Ncols \_do {\_unskip \_global\_setbox1##1=\_lastbox}}
1238     \_for numstep -2: \_Ncols..1 \_do {
1239       \_hrule height0pt\_kern5pt\_nobreak\_vfill
1240       \_ifnum\_Ncols=##1 \.botTitle{#1}[#2]\_else \.botTitle{}[]\_fi
1241       \_kern3pt \_nobreak
1242       \_hbox to\_hsize{%
1243         \_rlap{\_LightGrey \_vrule height\_tmpdim depth6pt width\_hsize}%
1244         \_kern\_parindent
1245         \_box1##1\_hss\_box1\_the\_numexpr##1-1
1246         \_kern\_parindent
1247       }
1248     } \_break
1249   }
1250 \.endbot
1251 }
1252 \_def\roundexpr#1{\_ea\_ea\_ea\roundexprA\_expr{#1}\_relax}
1253 \_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 register a new action by `\newaction{<full-vref>}{\dotopCite{<text>}}`.

op-bible.opm

```

1264 \_def\putCite #1 #2{% chap:verse {text}
1265   \_edef\fullvref{\_gentovref{#1}}%
1266   \_edef\fullvrefm{\_ea\renumvref\fullvref\_relax}%
1267   \_ea\newaction\_ea\fullvrefm{\dotopCite{#2}}%
1268 }
1269 \_nspublic \putCite ;

```

`\dotopCite {<text>}` creates the citation text by `\topinsert... \endinsert` from 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 `\sxddef{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.

```

1281 \_newcount\citenum
1282 \_def\dotopCite #1{%
1283   \_topinsert
1284   \_typosize[12/16]\_bi
1285   \_incr\citenum
1286   \_ifodd \_trycs{ct!\_the\citenum}{0}\_relax
1287   \_leftskip=.3\_hsize plus1fil \_parfillskip=0pt
1288   \_noindent
1289   \_rlap{\_hskip\_hsize \_kern-\_leftskip \_copy\lqqbox}\_hfill
1290 \_else
1291   \_let\quotedby=\_quotedbyright
1292   \_rightskip=.3\_hsize plus 1fil
1293   \_noindent \_llap{\_copy\lqqbox}%
1294 \_fi
1295 {\_printCite{#1}\_unskip}\_par
1296 \_ewref\_sxdef{{ct!\_the\citenum}{\_string\mypage}}%
1297 % \_vskip-.3\baselineskip
1298 \_endinsert
1299 }
1300 \_def\printCite#1{\_pdfliteral{2 Tr .15 w .9 g}#1\_pdfliteral{0 Tr 0 w 0 g}}
1301 \_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 quotation to the next line. The macro `\quotedbyright` (which is used at left pages) prints the `<author>` at the last line if there is sufficient space.

```

1311 \_newbox\lqqbox
1312 \_newbox\rqqbox
1313 \_setbox\lqqbox=\_hbox{\_lower3pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed,}}
1314 \_setbox\rqqbox=\_hbox{\_kern2pt\_lower38pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed}}
1315 \_ht\lqqbox=0pt \_dp\lqqbox=0pt
1316 \_ht\rqqbox=0pt \_dp\rqqbox=0pt
1317
1318 \_def\quotedby{\_par}
1319 \_def\quotedbyright#1{%
1320   \_unskip\_nobreak\_hfill\_penalty0\_hskip2em
1321   \_null\_nobreak\_hskip\_iindent\_hbox{#1}}

```

The following macros `\Cite`, `\insertCite` and `\swapCites` are used for insertion of citations to the two-column printed articles. The `\Cite<label>{<text>}` simply saves the `<text>` to the macro `\c!<article-num>{<label>}`. The `\insertCite<label>{<left-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>}`.

```

1335 \_def\Cite #1#2{\_sdef{c!\_the\articlenum!#1}{#2}}
1336 \_def\insertCite #1#2{\_def\citelabel{#1}%
1337   \_ifx\_left#2\insertCiteleft
1338   \_else \_ifx#2\_right\insertCiteright\_else
1339     \_errmessage{\_noexpand\insertCite#1: \_noexpand\left or \_noexpand\right expected}%
1340   \_fi\_fi
1341 }
1342 \_def\insertCiteleft {%
1343   \_ifnum\citepg=1 \_printwarn{\_noexpand\insertCite\citelabel: \_noexpand\swapCites activated}\_fi
1344   \_ifodd \_numexpr\_trycs{cp!\_the\articlenum!\citelabel}{0}+\citepg\_relax
1345   \_else \_insertCitelr \_left \_fi
1346 }
1347 \_def\insertCiteright{%
1348   \_ifodd \_numexpr\_trycs{cp!\_the\articlenum!\citelabel}{0}+\citepg\_relax
1349   \_insertCitelr \_right \_fi
1350 }
1351 \_def\insertCitelr#1{\_unskip\_vadjust{\_vbox{%
1352   \_ewref\_sxdef{{cp!\_the\articlenum!\citelabel}{\_string\mypage}}%
1353   \_vskip6pt
1354   \_advance\_hsize by\_parindent
1355   \_typosize[12/16]\_bi\Grey
1356   \_ifx#1\_left

```

```

1357     \_def\quotedby{\_par\_hfill}
1358     \_rightskip=\_parindent plus1fil \_leftskip=0pt
1359     \_setbox0\_vbox{%
1360         \_medskip \_noindent
1361         \_llap{\_copy\lqqbox}\_ignorespaces
1362         \_printCite{\_cs{c!\_the\articlenum!\.citelabel}}\_medskip}%
1363     \_hbox{\_kern-\_parindent\_rlap{White
1364         \_vrule height\_ht0 width\_hsize}\_box0}%
1365     \_else
1366         \_leftskip=\_parindent plus1fil
1367         \_parfillskip=0pt
1368         \_setbox0\_vbox{%
1369             \_medskip \_noindent
1370             \_rlap{\_hskip\_hsize\_kern-\_parindent\_copy\lqqbox}\_hfill
1371             \_ignorespaces \_printCite{\_cs{c!\_the\articlenum!\.citelabel}}\_medskip}%
1372         \_rlap{\_rlap{White \_vrule height\_ht0 width\_hsize}\_box0}%
1373     \_fi
1374     \_vskip6pt
1375 }}}
1376 \_def\swapCites{\_def\citepg{1}}
1377 \_def\citepg{0}
1378
1379 \_nspublic \Cite \insertCite ;

```

## Insertions into the intro text

op-bible.opm

```

1387 %% TBN page 236
1388
1389 \_newcount\shapenum
1390 \_newdimen\ii \_newdimen\w
1391 \_def\oblom #1 od #2 odsadit #3 {\_par \_ii=#1 \_w=\_hsize
1392     \_ifdim\ii>\_zo \_advance\w by-\_ii
1393     \_else \_advance\w by\_ii \_ii=\_zo \_fi
1394     \shapenum=1 \_tmpnum=0 \_def\shapelist{}
1395     \_loop \_ifnum\shapenum<#2 \_edef\shapelist{\shapelist\_zo\_hsize}%
1396         \_advance\shapenum by1 \_repeat
1397     \_loop \_edef\shapelist{\shapelist\ii\w}%
1398         \_advance\_tmpnum by1 \_ifnum\_tmpnum<#3 \_repeat
1399     \_advance\shapenum by#3 \_edef\shapelist{\shapelist\_zo\_hsize}
1400     \doshape}
1401 \_def\doshape{\_parshape \shapenum \shapelist}
1402 \_newcount\globpar
1403 \_ifx\_partokenset \_undefined \_def\partoken{\par} \_else \_def\partoken{\_par} \_fi
1404 \_def\doshape{\_global\globpar=0 \_ea\_def\partoken{\_ifhmode\shapepar\_fi}}
1405 \_def\shapepar{\_prevgraf=\_globpar \_parshape\shapenum\shapelist
1406     \_endgraf \_global\globpar=\_prevgraf
1407     \_ifnum \_prevgraf>\shapenum \_ea\_let\partoken=\_endgraf \_fi
1408 }
1409
1410 \_def\Citehereleft #1 (#2) #3{{
1411     \_par
1412         \_def\quotedby{\_par\_hfill}
1413         \_rightskip=\_parindent plus1fil \_leftskip=0pt
1414         \_setbox0\_vbox{%
1415             \_typosize[12/16]\_bi\Grey
1416             \_hsize=.5\_hsize
1417             \_medskip \_noindent
1418             \_llap{\_copy\lqqbox}\_ignorespaces
1419             \_printCite{#3}\_medskip}}%
1420     \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1421     \_xdef\lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1422     \_nointerlineskip\_vbox to0pt{\_kern#1\_baselineskip #2
1423         \_hbox{\_rlap{White
1424             \_kern-3mm\_vrule height\_ht0 width.5\_hsize}\_box0}%
1425     \_vss}}
1426     \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1427     \oblom {.5\_tmpdim} od #1 odsadit {\lines}
1428 }
1429 \_def\Citehereright #1 (#2) #3{{
1430     \_par

```

```

1431 \_def\quotedby{\_par\_parfillskip=0pt\_hfill}
1432 \_leftskip=\_parindent plusifill\_rightskip=0pt
1433 \_setbox0\_vbox{%
1434 \_typosize[12/16]\_bi\Grey
1435 \_hsize=.5\_hsize
1436 \_vskip\_medskipamount\_rlap{\_kern\_hsize\_copy\rqbox}\_vskip-\_medskipamount
1437 \_printCitef{\_noindent\_ignorespaces#3}\_medskip}}%
1438 \_tmpdim=\_ht0\_advance\_tmpdim by\_baselineskip
1439 \_xdef\_lines{\_the\_numexpr\_number\_tmpdim /\_number\_baselineskip\_relax}%
1440 \_nointerlineskip\_vbox toOpt{\_kern#1\_baselineskip #2
1441 \_hbox to\_hsize{\_hss
1442 \_llap{\White\_vrule height\_ht0 width.5\_hsize\_kern-3mm}%
1443 \_llap{\_box0}}
1444 \_vss}}
1445 \_tmpdim=\_hsize\_advance\_tmpdim by-2\_leftskip
1446 \_oblong {-0.5\_tmpdim} od #1 odsadit {\_lines}
1447 }
1448
1449 \_def\Citehere{\_par\_ifodd\_pageno\_ea\Citehereright\_else\_ea\Citehereleft\_fi}
1450
1451 \_nspublic\Citehere ;
1452
1453 \_def\insertBot #1#2[#3]#4(#5)#6{% {Title} [label] (params) {data}
1454 \_botinsert
1455 \_botTitle{#1}[#3]%
1456 \_kern3pt\_nobreak
1457 \_vbox{\_picwidth=\_hsize #5 #6}%
1458 \_endbot
1459 }
1460 \_def\putBot #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1461 \_edef\_fullvref{\_gentovref{#1}}%
1462 \_edef\_fullvrefm{\_ea\_renumvref\_fullvref\_relax}%
1463 \_ea\_newaction\_ea{\_fullvrefm}{\insertBot{#2}[#4](#6){#7}}%
1464 }
1465
1466 \_def\_c[#1/#2]#3{% text podel krivky: \c[init-rotace/repetice]{text}
1467 \_pdfsave\_pdfrotate{#1}\_rlap{\_edef\_tmpb{#3}\_replstring\_tmpb{ }{ } }\_def\_tmpa{#2}%
1468 \_ea\_foreach\_tmpb\_do{##1\_tmpa}}\_pdfrestore\_kern10mm
1469 }
1470 \_nspublic \insertBot \putBot \c ;

```

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

op-bible.opm

```

1479 \_def\.printintro{%
1480 \_begblock
1481 \_dest[i:\currbook/]
1482 \_chaptit{\_mtext{intro}}%
1483 \_input{\introfile}
1484 \_endblock
1485 }

```

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.

op-bible.opm

```

1493 \_newcount\.blocklevel % nesting level of blocks
1494 \_def\.begblock{\_par\_bgroup
1495 \_advance\.blocklevel by1\_advance\_leftskip by\_iindent\_rightskip=\_leftskip
1496 \_medskip
1497 \_pdfsavepos\_ea\_wref\_ea\Xblock\_ea{\_ea{\_the\.blocklevel}B{\_the\_pdflasttypos}}
1498 \_nobreak\_medskip
1499 }
1500 \_def\.endblock{\_par\_nobreak\_medskip
1501 \_pdfsavepos\_ea\_wref\_ea\Xblock\_ea{\_ea{\_the\.blocklevel}E{\_the\_pdflasttypos}}
1502 \_medskip\_egroup
1503 }
1504 \_refdecl{%
1505 \_def\Xblock#1#2#3{\_ifnum#1=1\_edef\_tmp{frm:\_ea\_ignoresecond\_currpage}~^J

```

```

1506 \_unless\_ifcsname \.tmp \_endcsname \_sxdef{\.tmp}{\_fi^^J
1507 \_sxdef{\.tmp}{\_cs{\.tmp}#2{#3}}\_fi}
1508 }
1509 \_newdimen\_.frtop \_newdimen\_.frbottom % positions of top and bottom text on the pages
1510 \_def\_.frcolor{.93 g } % light grey -- color of blocks.
1511 \_pgbackground={%
1512 \_slet{\_opb\_tmp}{frm:\_the\_gpageno}
1513 \_ifx\_.tmp\_undefined \_def\_.tmp{\\_fi
1514 \_.frtop=\_dimexpr \_pdfpageheight-\_voffset+\_smallskipamount\_relax
1515 \_.frbottom=\_dimexpr \_pdfpageheight-\_voffset-\_vsize-\_medskipamount\_relax
1516 \_ifx\_.frnext y \_edef\_.tmp{B{\_number\_.frtop}\.tmp}\_global\_let\_.frnext n\_fi
1517 \_ea\_.printframes \.tmp B{0}E{\_number\_.frbottom}
1518 \_ifx\_.frameslist\_empty \_else
1519 \_pdfliteral{q \_.frcolor 1 0 0 1 0 \_bp{-\_pdfpageheight} cm \.frameslist Q}\_fi
1520 }
1521 \_def\_.printframes B#1#2E#3{\_ifnum#1=0 \_else
1522 \_.printframe {\_hoffset}{#3sp}{\_xhsize}{\_ifnum#1=-1 \_number\_.frtop\_else#1\_fi sp-#3sp}
1523 \_ifx^#2\_else \_global\_let\_.frnext=y \_let\_.printframes=\_relax \_fi
1524 \_ea\_.printframes\_fi
1525 }
1526 \_def\_.frameslist{}
1527 \_def\_.printframe #1#2#3#4{\_edef\_.frameslist{\.frameslist
1528 \_bp{#1} \_bp{#2} \_bp{#3} \_bp{#4} re f }%
1529 }

```

## 13 Outline

op-bible.opm

```

1537 \_newdimen\_.colsep
1538 \.colsep=10pt
1539
1540 \_def\_.Outline{
1541 \_medskip
1542 % \_filbreak
1543 \_chaptit{\_mtext{outline}}}%
1544 \_everylist={\_ifcase\_ilevel \_or \_style I \_or \_style A \_or \_style n \_fi}
1545 \_sdef{\_item:A}{\_strut\_uppercase\_ea{\_athe\_itemnum}. }
1546 \_sdef{\_item:I}{\_strut\_uppercase\_ea{\_romannumeral\_itemnum}. }
1547 \_hsize=.5\_hsize \_advance\_hsize by-\_colsep
1548 \_emergencystretch=40pt
1549 \_leftskip=0pt \_rightskip=0pt
1550 }
1551 \_def\_.rightnote#1{\_par
1552 \_setbox0=\_hbox{\_kern\_hsize \_kern\_.colsep
1553 \_vtop{\_leftskip=0pt \_kern0pt\_noindent\_strut\_it#1}}
1554 \_ht0=0pt \_dp0=0pt \_box0 \_nointerlineskip
1555 }
1556 \_nspublic \_Outline \_rightnote ;

```

## 14 Typesetting variants

By default, chapter numbers are in the outer margin and quotes characters too. The `\normalchapnums` macro moves chapter 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.

op-bible.opm

```

1570 \_def\_.normalchapnumbers{
1571 \_margins/2 a2 (25,25,20,20)mm
1572 \_lrmargin=0pt
1573 \_setbox0=\_box\_.lqqbox \_setbox0=\_box\_.rqqbox
1574 \_def\_.printbeforefirst{%
1575 \_nobreak\_medskip
1576 \_.printchapnote
1577 \_hangindent=\_parindent \_hangafter=-2
1578 \_noindent \_llap{\_vbox to0pt
1579 {\_kern-8pt\_hbox{\_setfontsize{at23pt}\_bf\Red\_the\_.chapnum\_kern5pt}\_vss}}%
1580 }
1581 }
1582 \_nspublic \_normalchapnumbers ;

```

## 15 Checking syntax

op-bible.opm

```

1590 \_def\checksyntax#1 {%
1591   \_let\processbooks=\_relax
1592   \_ifx\_relax#1\_relax \_else
1593     \_begingroup
1594       \_the\syntaxmacros
1595       \_wterm{^^J** checking file: #1 **^^J}
1596       \_input{#1}
1597       \_vfil\_break
1598     \_endgroup
1599   \_ea\checksyntax \_fi
1600 }
1601
1602 \_newtoks\syntaxmacros
1603 {\_catcode\<=13
1604 \_global\syntaxmacros={
1605   \_def<#1>{\_bgroup
1606     \_message{checking \_unexpanded{<#1>}}%
1607     \_ifx\_relax#1\_relax \_errmessage{empty link}\_nobref\_else \_afterfi{\_checkbref#1>\_bref#1>}\_fi
1608     \_glet\linkpre=\_linkpre \_glet\linkfspec=\_linkfspec
1609   \_egroup
1610 }
1611 \_def\checkbref#1#2>{%
1612   \_isinlist{.#1#2}{<}\_iftrue \_errmessage{duplicated \_string<}\_nobref\_else
1613   \_ifx"#1\checkbrefQ #1#2>\_else \_checkbrefD #1#2>\_fi\_fi
1614 }
1615 \_def\checkbrefQ "#1"#2#3>{\_checkbrefD #2#3>}
1616 \_def\checkbrefD #1>{%
1617   \_isinlist{.#1}{ }\_iftrue\_checkbrefS#1>\_else\_checkbrefN#1>\_fi
1618 }
1619 \_def\checkbrefS #1 #2>{\_checkbrefN#2>}
1620 \_def\checkbrefN #1>{%
1621   \_def\tpmb{#1}
1622   \_ifx\tpmb\_empty \_errmessage{missing link data}\_nobref\_else
1623     \_replstring\tpmb{:}{ }\_replstring\tpmb{-}{ }\_replstring\tpmb{ }{ }%
1624     \_replstring\tpmb{a}{ }\_replstring\tpmb{b}{ }\_replstring\tpmb{c}{ }%
1625     \_setbox0=\_hbox{\_tmpnum=0\tpmb\_relax}%
1626     \_ifdim\_wd0>0pt \_errmessage{nonnumeric link data}\_nobref\_fi
1627   \_fi
1628 }
1629 \_def\_nobref{\_def\_bref##1>{\_Red\_string<##1>}}
1630 \_def\currbook{}
1631 \_def\prelinkB{BK}
1632 \_def\prelinkC{BK}
1633 \_def\prelinkV{0}
1634 \_def\nochapbooks{BK}
1635 \_let\<=<
1636
1637 \_def\x/#1/{\_def\tpmb{#1}%
1638   \_isinlist\tpmb{x}\_iftrue \_badx
1639   \_else \_isinlist\tpmb<\_iftrue \_badx
1640   \_else \_isinlist\tpm\enditems\_iftrue \_badx \_else \_x/#1/\_fi\_fi\_fi
1641 }
1642 \_def\_badx{\_errmessage{unclosed \_string\x/...}}
1643
1644 \_def\Article[#1]{}
1645 \_def\Cite #1 {\_par\_noindent{\_bf Cite: }}
1646 \_def\insertCite #1#2{}
1647
1648 \_def\putArticle #1 #2[#3]#4(#5){}
1649 \_def\putCite #1:#2 {\_par\_noindent{\_bf Cite: }}
1650 \_def\putBot #1 #2[#3]#4(#5){\_vbox}
1651
1652 \_def\c[#1/#2]#3{#3}
1653
1654 \_long\_ea\_def\_csname Note\_endcsname #1 #2#3%
1655
1656 {\_par \_let\_nextww\_undefined \_noindent{\_bf Note #1:} #3\_par}

```

```

1657 }}
1658 \nspublic \checksyntax ;

```

## 16 TODO macros

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

op-bible.opm

```

1668 \_def\chaptit#1{\_line{\_hss\chapfont\Red#1\_hss}
1669 \_nobreak
1670 }
1671 \_def\schaptit#1{\_bigskip\chaptit{#1}\_medskip}
1672
1673 \nspublic \chaptit ;
1674
1675 \_sdef{\_mt:intro:en}{Introduction} \_sdef{\_mt:outline:en}{Outline}
1676 \_sdef{\_mt:intro:cs}{Úvod} \_sdef{\_mt:outline:cs}{Osnova}
1677
1678 \_def\dopsat{{\Red !!! DOPSAT !!! }}
1679
1680 \_def\.\bibleinput#1 {\_bgroup
1681 \_catcode`##=13 \_bgroup\_lccode`~=# \_lowercase{\_egroup\_let~}=\.processline
1682 \_input{#1}%
1683 \_egroup
1684 }

```

Active character < used for references.

op-bible.opm

```

1690 \_def\_afterload{\_adef<{\.bref}}
1691 \_afterload
1692
1693 \_endnamespace

```

## 17 Index

\amark 2	\bpo! 2–3	\.newbook 2–3
\.begblock 19	\bpr! 2–3	\nochapbooks 3
\bex! 2–3	\.brefBookChapter 3	\notesfile 2
\.bibleinput 2	\.btit 2	\pbook! 2
\bibname 3	\btit! 3	\printedbooks 2
\bmark 2–3	\.checknochapbooks 3	\.printintro 19
\BookException 3	\.endblock 19	\.printwarn 1
\BookExceptions 2	\f! 3	\processbooks 2
\BookPost 2–3	\fmtfile 2	\.processbooks 3
\BookPre 2–3	\introfile 2, 19	\.sedef 1
\BookTile 3	\.myaddto 1	\.setheadline 3