

OpBible – Technical Documentation

Version: 0.31, 2023-10-04

The code of the `opbible.opm` macro file is described here. See also the user documentation in the file `opbible-doc.pdf`.

Table of contents

1	Preparatory work	1
2	Fonts	2
3	Usable macros	2
4	The main loop over Bible books	2
5	Book titles	4
6	Actions	5
7	The <code>\Note</code> macro	5
8	Inserting data from format files	9
9	Printing verses from <code>.txs</code> files	10
10	Bible references	11
11	Language variants	15
12	Inserting notes to the page	17
13	Inserting images and articles to the page	18
14	Inserting images over two pages	20
15	Inserting citations to the page	22
16	Chiasm	26
17	Outline	27
18	Timelines	27
19	Typesetting variants	28
20	Checking syntax	28
21	Generating templates from templates	29
22	Other macros	30
23	Setting active character and <code>\outer</code> macros	31
24	Index	31

1 Preparatory work

```
4 \_codedecl \processbooks {OpBible: macros for creating annotated Bible} opbible.opm
```

Printing version.

```
10 \_message{This is OP-Bible, version <\_opb_version>} opbible.opm
```

Loading packages.

```
16 \_load[vlna] % single-letter prepositions and splitting hyphen managed specially in Czech opbible.opm  
17 \_load[mte] % micro typographical extensions
```

Namespace of internal macros of `opbible`.

```
23 \_namespace{opb} opbible.opm
```

Basic settings of \TeX parameters.

```

29 \_newdimen\lrmargin \lrmargin=10mm
30 \_margins/2 a4 (23,27,20,20)mm
31
32 \_typosize[11/13] % typesetting size of Bible text
33 \_hyperlinks\Blue\Blue % hyperlinks activated
34
35 \_parindent=20pt
36 \_nopagenumbers
37 \_mte_enablemte % micro typographical extensions enabled
38 \_vlna_singlechars {Czech}{AaIiVvOoUuSsZzKk} % lowercase "a" added to this family
39
40 \_showboxbreadth=0
41 \_let\notecolor=\Red
42
43 \_def\LightGrey {\_setcmykcolor{0 0 0 .1}}
44 \_def\LiRed {\_setcmykcolor{0 .2 .2 0}}

```

2 Fonts

The Biblon font family has commercial license but it is very suitable for Bible typesetting. If it is present on your system, we use it. Otherwise, we use Termes font.

```

53 \_fontfam[lm]
54 \_fontfam[Heros] % fonts for notes
55 \_fontfam[biblon] % fonts for Bible text
56 \_ifx\Biblon\undefined % replace font if Biblon is unavailable:
57 \_fontfam[Termes]
58 \_let\Biblon=\Termes
59 \_fi
60
61 \_fontdef\bookfont{\_setfontsize{at19.pt}\_bf}
62 \_fontdef\chapfont{\_setfontsize{at13.pt}\_bf}
63 \_fontdef\markfont{\_setfontsize{at7pt}\_rm}
64 \_fontdef\captionfont{\Heros\cond\_setfontsize{at8pt}\_bf}
65 \_def\headfont{\Biblon\_setfontsize{at10pt}\_rm}
66 \_nsprivate \Biblon ;

```

3 Usable macros

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

```

77 \_let\printwarn=\opwarning
78 \_def \.sedef #1{\_ea\_edef \_csname#1\_endcsname}
79 \_long\_def\myaddto#1#2{\_ifcsize#1\_endcsize
80 \_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.

```

```

89 \_def\isspacein #1 #2\_iftrue{\_isempty{#2}\_iffalse}
90 \_def\iscolonin #1:#2\_iftrue{\_isempty{#2}\_iffalse}
91 \_def\isdivisin #1-#2\_iftrue{\_isempty{#2}\_iffalse}

```

4 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 $\langle a\text{-mark} \rangle$ (an actual mark of the book used in the text).
- Defines `\bmark` as $\langle b\text{-mark} \rangle$ (a mark of the book used in file names).
- Defines `\.btit` as the book title.
- Saves $\langle a\text{-mark} \rangle$ to the `\.currbook` macro.
- Calls `\.newbook{\langle a\text{-mark} \rangle}`
- Prints title of the book to the terminal and to the log.
- Calls `\bex!\langle a\text{-mark} \rangle` in order to apply the `\BookException` data.
- Inputs introduction file if it exists. The real `\input` and formatting 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!\langle a\text{-mark} \rangle` 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!\langle a\text{-mark} \rangle` 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 \TeX memory is freed.

opbible.opm

```

131 \_def\.processbooks {\_par
132   \_ifx\tmark\_undefined \_def\tmark{none}\_fi
133   \.checknochapbooks
134   \_useit{\_ea\.processbooksA \printedbooks} {}
135   \_useit{\_ea\.processbooksB \printedbooks} {}
136 }
137 \_def\.processbooksA #1 {%
138   \_if\_relax#1\_relax \_else \_sxddef{pbook!#1}{\_ea\.processbooksA \_fi
139 }
140 \_def\.processbooksB #1 {%
141   \_if\_relax#1\_relax \_else
142     \_edef\amark{#1}
143     \_edef\bmark{\_cs{f!#1}}
144     \_edef\.btit{\_cs{btit!#1}}
145     \_begingroup
146       \_edef\.currbook{#1}
147       \.newbook{#1}
148       \_wterm{^^J** \_cs{btit!#1} {#1} (\string\tmark: \tmark) **^^J}
149       \_cs{bex!#1}
150       \_isfile{\introfile}\_iftrue \.printintro
151       \_else \.printwarn{File with introduction text \introfile\_space not found}\_fi
152 %
153       \_isfile{\fmtfile}\_iftrue \_input{\fmtfile}
154       \_else \.printwarn{File with format info \fmtfile\_space not found}\_fi
155       \_isfile{\notesfile}\_iftrue \_input{\notesfile}
156       \_else \.printwarn{File with notes \notesfile\_space not found}\_fi
157       \_cs{bpr!#1}
158       \.bibleinput{\txsfile}
159       \.chapafter % material after the last chapter
160       \_cs{bpo!#1}
161     \_endgroup
162     \_ea \.processbooksB
163   \_fi
164 }
165 \_nspublic \processbooks ;

```

`\.newbook{\langle a\text{-mark} \rangle}` ejects previous page, prepares header and prints the book title.

opbible.opm

```

171 \_def\.newbook#1{\_vfil\_supereject
172   \_let\prelinkB=\.currbook \.chapnum=0
173   \_def\prelinkC{0}\_def\prelinkV{0}
174   \_global\_headline={\_hfil \_ea\setheadline\_ea{\.btit}}
175   \_line{\_hss\bookfont\.btit\_hss}
176   \_label[cref!#1]\_wlabel{#1}

```

```

177 \_par\_nobreak\_medskip
178 }

```

`\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.

opbible.opm

```

187 \_def\setheadline#1{\_global\_headline={\_headfont
188 \_ifodd\_pageno
189 \_rlap{\_it\bibname\_hss}%
190 \_hfil \_the\_pageno\_hfil
191 \_hbox to\lrmargin{\_hss\_bf#1\_ifx^\_botmark^\_else\_space \_botmark\_fi}%
192 \_kern-\lrmargin
193 \_else
194 \_kern-\lrmargin
195 \_hbox to\lrmargin{\_bf#1 \_firstmark\_hss}%
196 \_hfil \_the\_pageno\_hfil
197 \_llap{\_hss\_it\bibname}%
198 \_fi
199 }
200 }
201 \_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, moreover, it checks if the `\nochapbooks` is defined. If not, it prints warning.

opbible.opm

```

214 \_def\checknochapbooks {%
215 \_ifx\nochapbooks\_undefined
216 \_printwarn{\_noexpand\nochapbooks (boks without chapters) undefined.}%
217 \_def\nochapbooks{}%
218 \_else \_edef\nochapbooks{\_space\nochapbooks\_space}\_fi
219 }

```

5 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\bttit!<a-mark>{<title>}`, `\def\fb!<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>`.

opbible.opm

```

236 \_def\genbooks{}
237 \_def\BookTitle #1 #2 #3{%
238 \_sxdef{btit!#1}{#3}\_sxdef{f!#1}{#2}\_sxdef{fb!#2}{#1}%
239 \_addto\genbooks{#2 }%
240 }

```

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.

opbible.opm

```

252 \_outer\_long\_def\BookException #1 #2{\_myaddto{bex!#1}{#2}}
253 \_outer\_long\_def\BookPre #1 #2{\_myaddto{bpr!#1}{#2}}
254 \_outer\_long\_def\BookPost #1 #2{\_myaddto{bpo!#1}{#2}}
255
256 \_nspublic \BookTitle \BookException \BookPre \BookPost ;

```

The `\ChapterPre{<code>}` and `\ChapterPost{<code>}` inserts `<code>` before each chapter and after each chapter. The `<code>` is the same for each chapter, it does not vary depending on the Book or Chapter number.

```

264 \_long\_def\ChapterPre #1{\_def\chapbefore{#1}}
265 \_long\_def\ChapterPost #1{\_def\chapafter{#1}}
266
267 %\_outer\_def\ChapterPre {\ChapterPre}
268 %\_outer\_def\ChapterPost {\ChapterPost} % be done at the end of this file

```

6 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.

opbible.opm

```

288 \_def\.newaction#1#2{%
289   \_unless\_ifcscname alist!#1\_endcscname \_sdef{alist!#1}{\_fi
290   \_ea\_addto\_cscname alist!#1\_endcscname{#2}%
291 }

```

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`.

opbible.opm

```

302 \_def\.replpre#1#2#3{%
303   \_ifx^#2^\_def\.tmp{#1}{\_ea\_ea\_ea\_def\_ea\_ea\_ea\.buff\_ea\_ea\_ea{\_ea\.tmp\.buff}}%
304   \_else
305     \.replbuff{#2}{#1{#2}}{#3}%
306   \_fi
307 }

```

`\.replprepost`*<text>**<pre>**<post>**<fail>* searches *<text>* in `\.buff` and adds *<pre>* before and *<post>* after the *<text>*. If the *<text>* is not found then *<fail>* is executed. The `\.replprepost` is used by `\fmtins` (with empty *<pre>*) because we want to insert the *<post>* material directly.

The `\fmtkeep` uses `\.replprepost` with empty *<pre>* and *<post>* together.

opbible.opm

```

318 \_def\.replprepost#1#2#3#4{\.replbuff{#1}{#2#1#3}{#4}}

```

Both, `\.replpre` and `\.replprepost`, use `\.replbuff`*<what>**<whom>**<fail>* which replaces first occurrence of *<what>* by *<whom>* in `\.buff`. If *<what>* doesn't exist then `\.text` is defined as *<what>* and *<fail>* is executed.

opbible.opm

```

328 \_def\.replbuff #1#2#3{%
329   \_def\.replpredo##1#1##2\_end{%
330     \_ifx\_end##2\_end \_def\.text{#1}#3% <fail>
331     \_else \.replsave ##1#2##2\_end \_fi
332   }%
333   \_def\.replsave##1#1\_end{\_def\.buff{##1}}%
334   \_ea\.replpredo\.buff#1\_end
335 }

```

7 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>*.

opbible.opm

```

349 \_def\.gentovref#1{\.currbook/\.gentovrefA#1-\end}
350 \_def\.gentovrefA#1-#2\end{#1}

```

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

opbible.opm

```
357 \_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 below) 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.

opbible.opm

```
368 \_def\.transformword#1{%
369   \_ifcsname v!\tmark!#1\_endcsname \_lastnamedcs
370   \_else #1\_fi
371 }
```

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

`\Note` does exactly following:

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

This is done by `\.AddNote` $\{\langle full-vref \rangle\}$ $\{\langle note-num \rangle\}$ $\{\langle tword \rangle\}$.

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

opbible.opm

```
417 \_newcount\.notenumber
418 \_def\.Note #1 #2{%
419   \_edef\.fullvref{\.gentovref{#1}}%
420   \_ea\.isversezero\.fullvref\_iftrue
421   \_ea\.NoteB
422   \_else
423     \_incr\.notenumber
424     \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
425     \_def\.tmp{#1}\.sedef{notepre!\_the\.notenumber}{\_ea\.renumlabel\.fullvrefm\_relax}%
426     \_ifx\.nextww\_undefined
427       {\_def\.printwarn##1{\_xdef\.tword{\.transformword{#2}}}%
428       \_else \_xdef\.tword{\.nextww}\_fi
429       \_afterfi{\_isnextchar={\.NoteA}\.NoteA={}}}%
430   \_fi
431 }
432 \_def\.NoteA=#1#2% #2 separated by \par or \_par:
433
434 {%
435   \_sdef{notetext!\_the\.notenumber}{\_ignorespaces#2}%
```

```

436 \sedef{noteref!\_the\.notenum}{\.fullvrefm}%
437 \_ifx\.nextww\_undefined
438 \_ifx^#1^\_sdef{pword!\_the\.notenum\_ea}\_ea{\.tword}\_else \_sdef{pword!\_the\.notenum}{#1}\_fi
439 \_else
440 \_sdef{pword!\_the\.notenum\_ea}\_ea{\.nextwwA}%
441 \_let\.nextww=\_undefined \_let\.nextwwA=\_undefined
442 \_fi
443 \.reducetword
444 \_ea\.addNote\_expanded{\.fullvrefm}{\_the\.notenum}{\.tword}}%
445 }
446 \_def\.addNote#1#2#3{%
447 \_ifx^#3^\_tword is empty
448 \_edef\.tmp{\_cs{notepre!#2}}%
449 \_ea \.isdivisin\.tmp-\_iftrue
450 \.newaction{#1}{\.replpre{\.doNote{#2}}{}}%
451 \_else
452 \.newaction{#1}{\_addto\.prebuff{\.doCNote{#2}}}%
453 \_fi
454 \_else
455 \.newaction{#1}{\.replpre{\.doNote{#2}}{#3}{\.notefail{#2}}}%
456 \_fi
457 }
458 %\_outer\_def\Note{\.Note} % will be done at the end of this macro file

```

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

opbible.opm

```

467 \_def\.NoteB #1% #1 separated by \par or \_par
468 {%
469 {%
470 \_sdef{chapnote!\.fullvref}{\_ignorespaces#1}%
471 }
472 \_def\.isversezero#1/#2:#3\_iftrue{\_ifnum #3=0 }

```

`\.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 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 `\.isdivisin` and `\.iscolonin`.

opbible.opm

```

485 \_def\.renumlabel#1/#2\_relax{#2%
486 \_ea\.isdivisin\.tmp-\_iftrue --\_ea\.renumlabelA\.tmp\_relax#2\_relax \_fi
487 }
488 \_def\.renumlabelA#1:#2-#3\_relax#4:#5\_relax{%
489 \.iscolonin#3:\_iftrue #3\_else \_the\_numexpr#5+#3-#2\_relax \_fi
490 }

```

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.

opbible.opm

```

505 \_def\.notefail#1{%
506 \.printwarn{\_csstring\Note: \.currverse: The text "\_unexpanded\_ea{\.text}" not found}%
507 \.replpre{\.doNote{#1}}{}}% Note is registered with the beginning of the verse
508 }

```

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

The $\langle chapter \rangle$: $\langle verse \rangle$ is printed from `\notepre!` only if it differs from previous one, i.e. from `\.prevnotepre`. The $\langle pword \rangle$ is printed with uppercase first letter by `\.upcasefirst` and with appended dot, but the dot is not printed if the $\langle pword \rangle$ ends by `?` or `!` or `..`


```

520 \_def\prevnotepre{}
521 \_def\doNote#1#2{%
522   \_edef\tmpb{\_cs{notepre!#1}}%
523   \.notelog{\_space\_space \_csstring\Note \.tmpb\_space {#2}={\_cs{pword!#1}} (#1)}%
524   \.noteinsert{%
525     {\_bf \_ifx\prevnotepre\tmpb \_else \.tmpb \_enskip \_glet\prevnotepre=\tmpb \_fi
526     \.trymakedest{n:\_cs{noteref!#1}}%
527     \_edef\tmpb{\_csname pword!#1\_endcsname}%
528     \_ifx\tmpb\_empty \_else
529       \_addto\tmpb{.\_relax}\.punctpword
530       \_ea\upcasefirst \.tmpb\_space
531     \_fi
532   }% end of \bf
533   \_cs{notetext!#1}}%
534   {\notecolor#2}%
535 }
536 \_def\_printfnotemark{}
537 \_def\_textindent#1{\_noindent}

```

The $\langle pword \rangle$ is typically all lowercase. But we want to capitalize the first letter of the $\langle pword \rangle$ when printing by $\backslash\upcasefirst$. You can say $\backslashlet\upcasefirst=\relax$ if you don't want this feature.

```

547 \_def\upcasefirst #1{\_uppercase{#1}}

```

The dot is added to $\langle pword \rangle$ when it is printed. But if $\langle pword \rangle$ ends by ! or ? or . then the added dot is ugly. We have to correct it in the $\backslash\punctpword$ macro. Note that $\langle pword \rangle$ is saved to $\backslash\tmpb$.

```

555 \_def\punctpword{\_replstring\tmpb{!\_relax}{!}\_replstring\tmpb{?\_relax}{?}%
556   \_replstring\tmpb{.\_relax}{.}}

```

When $\backslash\Note$ has empty parameter $\langle word \rangle$ (i.e. $\langle tword \rangle$) then it is anchored to the beginning of the verse. Moreover, if there are more such Notes referenced to the same verse then we merge all such notes to single note. So $\backslash\doCNote\langle notenum \rangle$ is run from $\backslash\prebuff$ and it only adds the text of the note to the $\backslash\Cnotetext$ buffer. When $\backslash\prebuff$ is completed then $\backslash\printCnote$ prints the merged note.

```

567 \_def\doCNote #1{%
568   \_edef\tmpb{\_csname pword!#1\_endcsname}%
569   \.notelog{\_space\_space \_csstring\Note \.tmpb\_space }={\_cs{pword!#1}} (#1)}%
570   \_edef\prevnotepre{\_cs{notepre!#1}}%
571   \_ifx\tmpb\_empty \_else
572     \_addto\tmpb{.\_punctpword
573     \_edef\tmpb{\_noexpand\_bf \_ea\upcasefirst\tmpb\_noexpand-}}%
574     \_ea\_addto \_ea\Cnotetext \_ea{\tmpb}%
575   \_fi
576   \_ea\_ea\_ea\_addto\_ea\_ea\_ea\Cnotetext\_ea\_ea\_ea{\_csname notetext!#1\_endcsname}%
577 }
578 \_def\printCnote{%
579   \_ifx\Cnotetext\_empty \_else
580     \.noteinsert{%
581       {\_bf \_ea\nobook\currverse\_relax \.trymakedest{n:\currverse}} \Cnotetext
582     }%
583   \_fi
584 }
585 \_def\nobook #1/#2\_relax {#2} % only chapter:verse is printed

```

$\backslash\reducetword$ does nothing by default. But $\backslash\megrednotes$ re-defines it, so all $\backslash\Notes$ are referenced to the beginning of the verse and nothing is searched. The $\backslash\Notes$ with the same verse are merged in this case using $\backslash\doCNote$.

```

594 \_def\reducetword{}
595 \_def\mergednotes{\_def\reducetword{\_def\tword{}}}
596 \_nspublic \mergednotes ;

```

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

opbible.opm

```
609 \_let\.notelog=\_wlog
```

8 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`.

`\.addpre\macro{<text>}` adds the text to the macro before its original contents.

opbible.opm

```
626 \_def\.fmtpre#1#2{\.newaction{\.gentovref{#1}}{\_addto\.fmtprebuff{#2}}}
627 \_def\.fmtpreind#1#2{\.newaction{\.gentovref{#1}}{\.addpre\.preindbuff{#2}}}
628 \_def\.fmtadd#1#2{\.newaction{\.gentovref{#1}}{\_addto\.buff{\_empty#2}}}
629 \_def\.fmtins#1#2#3{\.newaction{\.gentovref{#1}}{\.replprepost{#2}}{\_empty#3}{\.fmtfail{#3}}}
630 \_def\.fmtfail#1{\.fmtwarn\_addto\.fmtprebuff{#1}}
631 \_def\.fmtwarn{\.printwarn{\_string\fmtins: \.currverse: The text "\.text" not found}}
632 \_def\.addpre#1#2{\_ea\.addpreA \_ea{#1}{#2}#1}
633 \_def\.addpreA #1#2#3{\_def#3{#2#1}}
634
635 \_nspublic \fmtpre \ftmadd \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 `\endcenter` because curious error (about # character) should be occur without this checking.

opbible.opm

```
644 \_newdimen\.centermargin \.centermargin=4em
645 \_def\.begcenter{\_par \_ifnum\_lastpenalty<10000 \_medskip \_fi
646 \_bgroup
647 \_def\.centeringmode{y}
648 \_parindent=0pt
649 \_leftskip=\.centermargin plusifill
650 \_rightskip=\_leftskip
651 }
652 \_def\.endcenter{\_par
653 \_ifx\.centeringmode\_undefined
654 \.printwarn{\_noexpand\endcenter ignored: no \_noexpand\begcenter precedes}
655 \_else \_egroup \_medskip \_fi
656 }
657 \_nspublic \begcenter \endcenter ;
```

`\ind{<number>}` gives an indentaion in the poetry environment. It is used in `\fmtpoetry`, the `\ind{<number>}` is inserted typically by `\fmtins` or `\fmtpre`. It ends the current line by `\par` only if we are not at beginning of a verse 1.

The `\spacefactor` is set to 1001, this value is used by the macro `\.hboxorllap`: the verse number is llaped after `\ind`.

opbible.opm

```
668 \_newifi\_ifpb_firstverse
669
670 \_def\.ind#1{\_unless \_ifpb_firstverse \_par \_else \_hskip-\_parindent \_fi
671 \_noindent
672 \_hskip#1\_iindent \_spacefactor=1001 \_ignorespaces}
```

`\fmtpoetry{<gen-vref>}{<fmt-data>}` saves `<gen-vref>` to `\.tmpa` and runs `<fmt-data>` in recursive loop using `\.fmtpoetA`. The `\.fmtpoetB` counts the number of slashes in local recursive loop and saves the result to the `_tmpnum`. The `\.fmtpoetC` inserts desired material using `\fmtprepoet` or `\fmtins` and using `\ind{_the_tmpnum}`.

opbible.opm

```
682 \_def\.fmtpoetry#1#2{\_def\.tmpa{#1}\.fmtpoetA #2\_end}
683 \_def\.fmtpoetA #1/{\_def\.tmpb{#1}\_tmpnum=1 \.fmtpoetB}
684 \_def\.fmtpoetB #1{\_ifx/#1 \_incr\_tmpnum \_ea\.fmtpoetB \_else \_afterfi{\.fmtpoetC#1}\_fi}
685 \_def\.fmtpoetC #1{%
686 \_expanded{\_ifx\.tmpb\_empty \_noexpand\.fmtpreind{\.tmpa}\_else
```

```

687 \noexpand\fmtins{\.tmpa}{\.tmpb}\_fi{ \noexpand\ind{\_the\_tmpnum}}}%
688 \_ifx\_end#1 \_else \_afterfi{\.fmtpoetA#1}\_fi
689 }
690 \_nspublic \ind \fmtpoetry ;

```

\fmtfont $\langle gen-vref \rangle$ $\langle whar \rangle$ $\langle cmd \rangle$ replaces $\langle what \rangle$ by $\backslash bgroup \langle cmd \rangle \langle what \rangle \backslash egroup$.
\fmtkeep $\langle gen-vref \rangle$ $\langle what \rangle$ replaces $\langle what \rangle$ by $\{ \langle what \rangle \}$, so it is unsearchable.
\fmtrepl $\langle gen-vref \rangle$ $\langle what \rangle$ $\langle wham \rangle$ replaces $\langle what \rangle$ by $\langle whom \rangle$.

opbible.opm

```

701 \_def\fmtfont#1#2#3{%
702 \_newaction{\.gentovref{#1}}{\.replprepost{#2}\bgroup#3}\egroup{\.fmtwarnf\fmtfont}}
703 \_def\fmtkeep#1#2{%
704 \_newaction{\.gentovref{#1}}{\.replpre{#2}\.fmtwarnf\fmtkeep}}
705 \_def\fmtrepl#1#2#3{\.newaction{\.gentovref{#1}}{\.replbuff{#2}{#3}\.fmtwarnf\fmtkeep}}
706
707 \_def\fmtwarnf#1{\.printwarn{\_string#1: \.currverse: The text "\.text" not found}}
708 \_nspublic \fmtfont \fmtkeep \fmtrepl ;

```

9 Printing verses from .txs files

When Bible text is processed then book mark is saved to $\backslash currbook$ and each input line is separated to the $\langle chapter-num \rangle$: $\langle verse-num \rangle$ and $\langle verse-text \rangle$.

The $\backslash processline \langle chapter \rangle$: $\langle verse \rangle$ $\langle space \rangle$ $\langle verse-text \rangle$ ^^J is repeatedly processed.

opbible.opm

```

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

```

$\backslash processverse \langle full-vref \rangle$ $\langle space \rangle$ $\langle verse-text \rangle$ $\backslash end$ does

- defines $\backslash currverse$ as $\langle full-vref \rangle$,
- prepares $\backslash currversenum$, $\backslash currversetext$, $\backslash currchapnum$ from $\langle full-vref \rangle$,
- defines $\backslash buff$ as $\langle verse-text \rangle$,
- processes all actions from $\backslash alist!$ $\langle full-vref \rangle$,
- if $\backslash currchapnum$ changed, prints $\backslash chapafter$ (for previous chapter) and $\backslash chapbefore$ (for new chapter).
- prints verse from $\backslash buff$ using $\backslash printverse$

opbible.opm

```

736 \_newcount\chapnum
737 \_def\processverse #1 #2\_end{%
738 \_xdef\currverse{#1}%
739 \_preparechapverse #1
740 \_let\prelinkV=\currversenum
741 \_gdef\buff{#2}\_gdef\fmtprebuff{\}_\_gdef\preindbuff{\}_\_gdef\prebuff{\}_\_gdef\Cnotetext{\}%
742 \_ifx\verseto\_empty \_csname alist!#1\_endcsname \_else
743 \_for num \versefrom..\verseto \_do{\_csname alist!\currbook/\currchapnum:#1\_endcsname}%
744 \_fi
745 \_ifnum\currchapnum=\chapnum \_else
746 \_ifnum\chapnum>1 \chapafter \_fi
747 \_let\prelinkC=\currchapnum \chapnum=\currchapnum\_relax
748 \chapbefore
749 \_label[cref!\currbook\_space\_the\chapnum]\_wlabel{\currbook-\_the\chapnum}%
750 \_fi
751 \printverse
752 }
753 \_def\preparechapverse #1/#2:#3 {\_def\currchapnum{#2}%
754 \_def\verseto{\}%
755 \_isdivisin #3-\_iftrue \.defversefromto #3\_end
756 \_else \_def\currversenum{#3}\_glet\currversetext=\currversenum
757 \_fi
758 }
759 \_def\defversefromto #1-#2\_end{%
760 \_def\versefrom{#1}\_def\verseto{#2}%
761 \_def\currversenum{#1}\_gdef\currversetext{#1--#2}}

```

User can do little changes in the verse text using $\backslash cnvtext \langle what \rangle$ $\langle replaced \rangle$. For example you can do $\backslash cnvtext \{ \} \{ \backslash bgroup \it \} \backslash cnvtext \{ \} \{ \} \backslash / \backslash egroup$ for making [words] in brackets printed italics.

```

769 \_def\prepareversetext{}
770 \_def\cnvtext#1#2{\_addto\prepareversetext{\_replstring\buff{#1}{#2}}
771 \_nspublic \cnvtext ;

```

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

`\.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`).

The `\.fmtprebuf` includes `\ind` command from `\fmtpoetry` if the verse should be indented at its begin before the verse number. The verse number is shifted up and it is in an `\hbox` or it is lapped in the poetry environment, more exactly immediately after `\ind` is used. The `\.hboxorllap` macro does this game.

```

787 \_def\.printverse{%
788   \fmtprebuff % material accumulated by \fmtpre
789   \_ifnum\.currversenum=1 \_firstversetrue \.printbeforefirst \_fi
790   \_quitmode \_mark{\.currchapnum:\.currversetext}%
791   \_ifx\verseto\_empty \_trymakedest{v:\.currverse}%
792   \_else \_fornum \versefrom..\verseto \_do{%
793     \_wlog{xxxxx v:\.currbook/\.currchapnum:##1}\_trymakedest{v:\.currbook/\.currchapnum:##1}%
794     \_fi
795     \preindbuff
796     \_raise5pt\.hboxorllap{\_unless\_ifnum\.currversenum=1 \_markfont\.currversetext\,\_fi}%
797     \_firstversefalse
798     \prepareversetext
799     \prebuff\.printCnote\buff \_space
800 }
801 \_def\.hboxorllap{\_ifnum\_spacefactor=1001 \_ea\_llap \_else \_ea\_hbox \_fi}
802
803 \_def\.printbeforefirst{%
804   \_par\_nobreak \_medskip
805   \_trychapnote
806   \_setbox0=\_vtop{\_kern-1.5ex \_ewref\_sxdef{{ch!\.currbook/\_the\chapnum}{\_string\mypage}}
807     \_hbox{\_setfontsize{at50pt}\_bf\LiRed\_the\chapnum}}
808   \_dp0=0pt
809   \_tmpdim=\_lrmargin
810   \_advance\_tmpdim by4pt
811   \_ifnum\_the\chapnum>9 \_advance\_tmpdim by19pt \_fi
812   \_ifodd\_trys{ch!\.currbook/\_the\chapnum}{0}
813     \_moveright\_tmpdim \_line{\_hss\_box0}
814   \_else \_moveleft\_tmpdim \_box0 \_fi
815   \_nobreak \_vskip-\_medskipamount
816   \_nobreak \_nointerlineskip \_noindent
817 }

```

`\.printchapnote{<text>}` implements printing the notes declared by `\Note <chapnum>:0`. It is run using `_trychapnote` only if the relevant not is declared.

```

824 \_def\.trychapnote{%
825   \_ifcscname chapnote!\.currbook/\_the\chapnum:0\_endcscname
826   \_printchapnote{\_cs{chapnote!\.currbook/\_the\chapnum:0}}\_fi
827 }
828 \_def\.printchapnote #1{\_par
829   {\_leftskip=\_parindent plus1fill \_rightskip=\_leftskip \_noindent\_it #1\_par}
830   \_medskip
831 }
832 \_nspublic \printchapnote ;

```

`\.chapbefore` is processed before each chapter. `\.chapters` is processed after each chapter. User can define values by `\ChapterPre` and `\ChapterPost` macros.

```

840 \_def\.chapbefore{\_bigskip} \_def\.chapters{}

```

10 Bible references

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`.

opbible.opm

```

868 \_def\.linktext{\.ltextP\.ltextB\.ltextC\.ltextV\.ltextS\.ltextF\.ltextN}
869 \_def\.bref #1>{\_let\.brefH=\_relax \_def\.linkspec{#1}\_isnextchar{"\.brefA"}{\.brefA"}#1>}
870 \_def\.brefA"#1" {\_def\.ltextP{#1}%
871 \_isnextchar { }{\_addto\.ltextP{~}\_afterassignment\.brefB\_let\.next= }%
872 {\_isnextchar_{ }{\_def\.brefH}{\_afterassignment\.brefB\_let\.next= }{\.brefB}}%
873 }
874 \_def\.brefB #1>{% #1 is link-spec
875 \_def\.ltextB{}\_def\.ltextC{}\_def\.ltextF{}\_def\.ltextN{}%
876 \_isspacein #1 \_iftrue
877 \_iscolonin #1:\_iftrue \.brefBookChapterVerse #1>%
878 \_else \.brefBookChapter #1>\_fi
879 \_else \_iscolonin #1:\_iftrue \.brefChapterVerse #1>%
880 \_else \.brefVerse #1>%
881 \_fi\_fi
882 \_def\.linkpre{v}%
883 \_isnextchar n{\_def\.linkpre{n}\.brefC}%
884 {\_isnextchar g{\_def\.linkpre{g}\.brefC}%
885 {\_isnextchar a{\_def\.linkpre{a}\.brefC}%
886 {\_isnextchar i{\_def\.linkpre{i}\.brefC}{\.brefD}}}}%
887 }
888 \_def\.brefC{\_afterassignment\.brefD \_let\.next= }
889
890 \_def\.brefBookChapterVerse #1 #2:#3>{\_def\.ltextB{#1~}\.brefChapterVerse #2:#3>}
891 \_def\.brefBookChapter #1 #2>{\_def\.ltextB{#1~}%
892 \_isinlist\nochapbooks{ #1 }\_iftrue
893 \_def\.ltextC{}\_let\.ltextCin=\.ltextnCin \_afterfi{\.brefVerse #2>}%
894 \_else \_afterfi{\.brefChapter #2>}\_fi}
895 \_def\.brefChapterVerse #1:#2>{\_def\.ltextC{#1:}\.brefVerse #2>}
896 \_def\.brefVerse #1>{%
897 \_isdivisin #1-\_iftrue \.brefFromTo #1>%
898 \_else \.versedef#1\_relax\_fi
899 }
900 \_def\.brefChapter #1>{%
901 \_isdivisin #1-\_iftrue \.brefFromTo #1>\_let\.ltextC=\.ltextV
902 \_else \_def\.ltextC{#1}\_fi
903 \_def\.ltextV{}\_def\.ltextS{}%
904 }
905 \_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.

opbible.opm

```

913 \_def\.versedef {\_afterassignment\.versedefB \_tmpnum=0}
914 \_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.

opbible.opm

```

921 \_def\.brefD{%
922 \_ifnum 0\.ltextV=0 \_def\.ltextV{}\_fi
923 \_if a\.linkpre \_ifx\.ltextV\_empty \_else \_edef\.ltextC{\.ltextV:}\_def\.ltextV{}\_fi\_fi
924 \_edef\.linkfspec{\_ea\.ltextBin\.ltextB~/\_ea\.ltextCin\.ltextC:/\_ea\.ltextVin\.ltextV:/}%
925 \.brefL
926 }
927 \_def\.ltextBin #1-#2/{\_ifx^#1^\.prelinkB \_else #1\_immediateassignment\_def\.prelinkB{#1}\_fi/}

```

```

928 \_def\ltextCin #1:#2/{\_ifx^#1~\prelinkC \_else #1\_immediateassignment\_def\prelinkC{#1}\_fi:}
929 \_def\ltextVin #1:#2/{\_ifx^#1~\prelinkV \_else #1\_immediateassignment\_def\prelinkV{#1}\_fi}
930 \_def\ltextnCin #1:#2/{\prelinkC:\_immediateassignment\_let\ltextCin=\ltextsCin}
931 \_let\ltextsCin=\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.

```

941 \_def\<{\_let\prelinkB=\currbook \_let\prelinkC=\currchapnum \_let\prelinkV=\currversenum \bref}

```

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: $\langle book \rangle /$.

If the book mark is declared by `\vdef` then the printed version of the book mark is transformed depending on the current `\tmark`. This is done by the the `\newlinkB` macro.

`\linklog` $\langle \{text\} \rangle$ 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.

```

965 \_def\brefL{%
966   \_edef\linkfspecm{\_ea\renumvref\linkfspec\relax}%
967   \_ifx\linkfspec\linkfspecm \_else
968     \_ea\_ea\_ea\renumlinktext \_ea\linkfspec \_ea\_relax \linkfspecm \relax
969     \_let\linkfspec=\linkfspecm
970   \_fi
971   \_ifx\ltextV\_empty \_ifx\ltextC\_empty \_else \_ea\linkfspecone \linkfspec\_end \_fi\_fi
972   \_if a\linkpre\_relax \_ea\linkfspecarticle \linkfspec\_end \_fi
973   \_if i\linkpre\_relax \_ea\linkfspecintro \linkfspec\_end \_fi
974   \_ifx \ltextB\_empty \_else \_ea \newltextB \ltextB \_fi
975 %   \message{\meaning\linkfspec, \meaning\ltextC, \meaning\currchapnum}%
976   \reducelinktext
977   \linklog{\_sspace <\_unexpanded\_ea{\linkspec}>\linkpost = [\linkpre:\linkspec]%
978     {\_ifx\brefH\_empty \ltextP \_else \linktext\_fi}}%
979   \ensuredest \createlink
980 }
981 \_def\linkfspecone #1:#2\_end {\_def\linkspec{#1:1}\_def\prelinkV{1}}
982 \_def\linkfspecarticle #1/#2:#3\_end {\_def\linkspec{#1/#2}}
983 \_def\linkfspecintro #1/#2\_end {\_def\linkspec{#1/}}
984
985 \_def\renumlinktext #1/#2:#3\_relax #4/#5:#6\_relax{%
986   \_ifx\ltextC\_empty \_else \_def\ltextC{#5}\_fi
987   \_def\ltextV{#6}%
988   \_ifx\ltextN\_empty \_else
989     \_ifx\ltextF\ltextDD
990       \_isinlist\ltextN{:}\_iftrue
991         \_ifcsname rn!\tmark!#1/\ltextN\_endcsname \_edef\ltextN{\_cs{rn!\tmark!#1/\ltextN}}%
992         \_fi
993       \_else \_edef\ltextN{\_the\_numexpr#6+\ltextN-#3\_relax}\_fi
994       \_else \_let\ltextN=\ignomit % \ltextN is a list of verses, for example 7,9,13
995         \_ea\_foreach\ltextN,\_do ##1,{\_edef\ltextN{\_tmp,\_the\_numexpr#6+##1-#3}}%
996         \_let\ltextN=\tmp
997       \_fi
998   \_fi
999 }
1000 \_def\ltextDD{--}
1001
1002 \_def\newltextB #1-{\_edef\ltextB{\_trycs{v!\tmark!#1}{#1}~}}
1003
1004 \_def\_sspace{\_space\_space\_space\_space}
1005 \_def\linkpost{\_if v\linkpre \_else \linkpre\_fi \_space}

```

`\.reducelinktext` does nothing or reduces printed link if its book is equal to the current book and if its chapter is equal to printed chapter. It is activated by `\reduceref` and deactivated by `\noreduceref`. The `\re` macro activates `\.reducelinktext` only for single `\.brief`.

opbible.opm

```

1014 \_def\.reducelinktextA{%
1015   \_edef\.tmp{\.currbook~}%
1016   \_ifx\.ltextB\.tmp \_def\.ltextB{%
1017     \_edef\.tmp{\_trycs{\_opb_currchapnum}{?}:}%
1018     \_ifx\.ltextC\.tmp \_def\.ltextC{%
1019       \_fi\_fi
1020       \_ifcsname \_opb_reA\_endcsname \_let\.reA=\.reA \_fi % after \re
1021   }
1022 \_def\.reduceref{\_let\.reA=\.reA \_let\.reA=\.reA}
1023 \_def\.noreduceref{\_let\.reA=\.reA \_let\.reA=\_relax}
1024 \_noreduceref % default
1025
1026 \_def\.re{\_let\.reA=\.reA \_let\.reA=\.reA}
1027
1028 \_nspublic \reduceref \noreduceref \re ;

```

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

opbible.opm

```

1034 \_def\tracinglinks{\_let\.linklog=\_wlog}
1035 \_def\notracinglinks{\_let\.linklog=\_ignoreit}
1036 \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.

`\tracingouterlinks` activates logging of broken links to non-existent books. By default, these links are not logged because we assume that no whole Bible is processed but only selected books.

opbible.opm

```

1050 \_def\.createlink{%
1051   \_ifx\.briefH\_empty \_let\.linktext=\.ltextP\_fi
1052   \_ea\.isprintedbook\.linkfspec \_iftrue
1053   \_link[\.linkpre:\.linkfspec]{\_ilinkcolor}{\.linktext}%
1054   \_else {\_ilinkcolor\.linktext}\_fi}%
1055 }
1056 \_def\.isprintedbook #1/#2\_iftrue{\_ifcsname pbook!#1\_endcsname}
1057 \_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 `.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 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 creates correct hyperlinks and page numbers.

opbible.opm

```

1077 \_newwrite\.xrf
1078 \_immediate\_openout\.xrf=\_jobname.xrf
1079 \_openref
1080
1081 \_def\.ensuredest{\_immediate\_write\.xrf{\_string\_sdef{\.linkpre:\.linkfspec}{}}}
1082 \_refdecl{
1083   \_isfile{\_jobname.xrf}\_iftrue \_input{\_jobname.xrf}\_fi^^J
1084   \_def\.Xdest#1{\_ifcsname pg:#1\_endcsname \_sdef{pg:#1}{\_ea\_usesecond\_currpage}\_fi}^^J
1085   \_def\.mypage{\_ea\_usesecond\_currpage}
1086 }
1087 \_def\.trymakedest#1{%
1088   \_ifcsname #1\_endcsname \_dest[#1]\_ea\_glet\_csname #1\_endcsname \_undefined \_fi
1089   \_ewref\.Xdest{#1}%
1090 }

```


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.

```

1098 \def\pg{%
1099   \ifcsname pg:\linkpre:\linkfspec\endcsname
1100     {\edef\linktext{\cs{pg:\linkpre:\linkfspec}}\let\brefH=\relax \createlink}%
1101   \else {\Red ??}\fi
1102   \immediate\write\xrf{\string\sdef{pg:\linkpre:\linkfspec}{??}}%
1103 }
1104 \nspublic \pg ;

```

`\cref` if simply `\ref` with `cref!` prefix.

```

1110 \def\cref[#1]{\ref[cref!#1]}
1111
1112 \nspublic \cref ;

```

11 Language variants

`\variants` \langle number-of-variants \rangle $\{\langle$ mark-A $\rangle\}$ $\{\langle$ mark-B $\rangle\}$ $\{\langle$ mark-C $\rangle\}$...

sets `\numvariants`= \langle number-of-variants \rangle and does `\def\tmarkA{\langlemark-A $\rangle}$` `\def\var!1{\langlemarkA $\rangle}$` `\def\var!2{\langlemark-B $\rangle}$` `\def\var!3{\langlemark-C $\rangle}$` etc.

```

1124 \newcount\numvariants
1125 \def\variants{\tmpnum=0 \afterassignment\variantsA \numvariants}
1126 \def\variantsA{%
1127   \ifnum\tmpnum<\numvariants
1128     \advance\tmpnum by1
1129     \afterfi{\variantsB{\the\tmpnum}}%
1130   \fi
1131 }
1132 \def\variantsB#1#2{%
1133   \ifnum#1=1 \gdef\tmarkA{#2}\sdef{var!1}{#2}%
1134   \else \sdef{var!#1}{#2}%
1135   \fi
1136   \variantsA
1137 }
1138 \nspublic \variants ;

```

`\vdef` $\{\langle$ phrase-A $\rangle\}$ $\{\langle$ phrase-B $\rangle\}$ $\{\langle$ phrase-C $\rangle\}$... does

`\def\v!\langlemark-B $\rangle!\langle$ phrase-A $\rangle\{\langle$ phrase-B $\rangle\}$` `\def\v!\langlemark-C $\rangle!\langle$ phrase-A $\rangle\{\langle$ phrase-C $\rangle\}$` 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{\langlenumber $\rangle\{\langle$ param $\rangle\}$` does real work and it defines (roughly sepaking):

```

If \langle param \rangle is " \def \v!\langle tmark \rangle!\langle phrase-A \rangle \{\langle previous param \rangle\}
else \def \v!\langle tmark \rangle!\langle phrase-A \rangle \{\langle param \rangle\}

```

```

1155 \def\vdef#1{\def\tmp{#1}%
1156   \ifcsname v!\_trycs{var!2}{!}\tmp\endcsname
1157     \printwarn{\noexpand\vdef used secondly for phrase {\tmp}, ignored}\fi
1158   \tmpnum=1 \ea\vdefA
1159 }
1160 \def\vdefA{%
1161   \ifnum\tmpnum<\numvariants
1162     \advance\tmpnum by1
1163     \afterfi{\.vdefB{\the\tmpnum}}%
1164   \fi
1165 }
1166 \def\vdefB#1#2{\def\tmpa{}}
1167 \ifx\vdef#2\def\tmpa{#2}\fi
1168 \ifx\tmpa\_empty
1169   \ifx^#2\_else
1170     \unless \ifcsname v!\_cs{var!#1}!\tmp\endcsname
1171       \sedef{v!\_cs{var!#1}!\tmp}{\_ifx^#2\prevcs{#1}\tmp \_else#2\_fi}%
1172     \fi\_fi
1173   \ea\vdefA

```



```

1174 \_else \_errmsgage{\_string\vdef: too few parameters. To be read again: \_string#2}%
1175 \_ea\_.tmpa
1176 \_fi
1177 }
1178 \_def\_.prevcs #1#2{\_ifnum#1=2 #2\_else \_cs{v!\_cs{var!\_the\_numexpr#1-1\_relax}!#2}\_fi}
1179
1180 \_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⟩`.

opibble.opm

```

1193 \_def\_.x/#1/{\_tryscs{v!\tmark!#1}{\xA#1/}}
1194 \_def\_.xA#1/{#1\_ifx\tmarkA\_undefined \_else \_ifx\tmark\tmarkA \_else
1195 \_printwarn{\_string\x/#1/ -- this phrase is undefined by \_csstring\vdef}%
1196 \_fi\_fi
1197 }
1198 \_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.

opibble.opm

```

1211 \_def\_.ww{%
1212 \_ifx\_.varnum\_undefined \_setvarnum \_fi
1213 \_tmpnum=0
1214 \_ifx\_.nextww\_undefined \_ea\_.wwA
1215 \_else \_printwarn{Only single \_csstring\ww must be before \_csstring\Note}%
1216 \_ea\_.wwB \_fi
1217 }
1218 \_def\_.wwA#1#2 {\_advance\_tmpnum by1
1219 \_isequal{"}{#1}\_iffalse
1220 \_def\_.nextww{#1}\_def\_.nextwwA{#2}%
1221 \_ifx\_.nextwwA\_empty \_let\_.nextwwA=\.nextww \_else \_ea \_redefwwA #2\_end \_fi
1222 \_fi
1223 \_ifnum\_.varnum=\_tmpnum \_ifnum\_tmpnum<\.numvariants \_ea\_ea\_ea \_.wwB \_fi
1224 \_else \_ea \_.wwA \_fi
1225 }
1226 \_def\_.wwB#1 {\_advance\_tmpnum by1
1227 \_ifnum\_tmpnum<\.numvariants \_ea\_.wwB \_fi
1228 }
1229 \_def\_.redefwwA =#1\_end{\_def\_.nextwwA{#1}}
1230
1231 % \_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` and more macros are defined as `\outer` in order to better diagnose mistakes with their parameters. But we want to skip such objects in `\switch` parameters. This is the reason why we set `_suppressoutererror=1` during the `\switch` is processed.

opibble.opm

```

1245 \_newtoks\_.switchD
1246 \_def\_.switch {\_let\_.switchN=\.switchA \_suppressoutererror=1 \.switchN}
1247 \_long\_def\_.switchA #1#2{\.switchD={#2\_let\_.switchN=\.switchI}%
1248 \_ifx\_relax#1\_relax \_the\_.switchD
1249 \_else \_foreach #1,\_do ##1,{\_def\tmp{##1}\.switchC}%
1250 \_fi
1251 \_futurelet\_.next\_.switchB
1252 }
1253 \_def\_.switchB{\_ifx\_.next\_bgroup \_ea\_.switchN \_else \_suppressoutererror=0 \_fi}
1254 \_long\_def\_.switchI #1#2{\_futurelet\_.next\_.switchB}

```

```

1255 \_def\switchC{\_ifx\tmp\tmark \_the\switchD \_fi}
1256
1257 \_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.

opbible.opm

```

1265 \_def\.setvarnum{\_gdef\.varnum{0}%
1266 \_ifnum\numvariants=0 \_gdef\.varnum{1}\_wlog{There is only single language variant (1)}%
1267 \_else
1268 \_tmpnum=0
1269 \_loop
1270 \_advance\_tmpnum by1
1271 \_ea\_ifx \_csname var!\_the\_tmpnum\_endcsname \tmark \_xdef\.varnum{\_the\_tmpnum}\_fi
1272 \_ifnum\_tmpnum<\.numvariants \_repeat
1273 \_ifnum \.varnum=0 \_errmessage{\_noexpand\tmark isn't set, \_noexpand\.setvarnum failed}%
1274 \_else \_wlog{Language variant set by \_string\tmark{\tmark} (\.varnum)}\_fi
1275 \_fi
1276 }

```

`\renum` $\langle book\text{-}mark \rangle \langle chapter\text{-}num \rangle : \langle verse\text{-}num \rangle = \langle t\text{-}mark \rangle \langle chap\text{-}num \rangle : \langle from \rangle - \langle to \rangle$ 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>}

```

opbible.opm

```

1290 \_def\.renum #1 #2:#3 = #4 #5:#6-#7 {%
1291 \_tmpnum=#3\_relax
1292 \_fornum #6..#7 \_do {\_sxdef{rn!#4!#1/#2:\_the\_tmpnum}{#5:#1}\_incr\_tmpnum}%
1293 }
1294 \_nspublic \renum ;

```

12 Inserting notes to the page

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

opbible.opm

```

1303 \_newinsert \.noteins
1304 \_skip\.noteins=\_bigskipamount % noterule height
1305 \_count\.noteins=500 % two columns
1306 \_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 $\langle text \rangle$ 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.

opbible.opm

```

1319 \_def\.noteinsert #1{\_insert\.noteins{%
1320 \.noteset
1321 \_vbox to\_ht\_strutbox{\_nobreak \_vskip-\_baselineskip
1322 #1\_unskip\_par \_nobreak \_vskip-\_baselineskip
1323 \_hbox{\_lower\_dp\_strutbox\_vbox{}}
1324 \_penalty0
1325 }}
1326 \_def\.noteset{Heros\cond \_scalemain \_typoscale[800/800] % Heros condensed 80%
1327 \_Black \_nobreak
1328 \_widowpenalty=20 \_clubpenalty=20
1329 \_leftskip=0pt \_rightskip=0pt \_parfillskip=0pt plusifill
1330 \_parindent=0pt
1331 \_lineskiplimit=-3pt
1332 \_hsize=.5\_hsize \_advance\_hsize by-1em\_relax % two columns
1333 \_everypar{}
1334 }

```

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).

```

1355 \_addto\_pagecontents{%
1356   \_ifvoid\.noteins \_else
1357     \_vskip\_skip\.noteins \noterule
1358     \_setbox\.noteins=\_vbox{\_penalty0 \_unvbox\.noteins \_vfil}
1359     \_splittopskip=12pt
1360     \_setbox0=\_vsplit\.noteins to0pt % adding \splittopskip to \.noteins
1361     \_def\_Ncols{2}
1362     \_dimen0=.5\_ht\.noteins \_setbox6=\_box\.noteins
1363     \_vskip\_splittopskip
1364     \_balancecolumns
1365   \_fi
1366   \_unless\_ifvoid\.botins \_unvbox\.botins
1367   \_else \_vskip 0pt plus1filll minus8pt \_fi
1368 }
1369 \_def \noterule {\_kern-3pt {\Black \_hrule width\_hsize}\_kern 2.6pt }
```

13 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.

```

1381 \_newinsert\.botins
1382 \_def\.botinsert{\_setbox0=\_vbox\_bgroup}
1383 \_def\.endbot{\_par\_egroup
1384   \_insert\.botins{\_splittopskip=0pt \_penalty100
1385   \_hrule height0pt \_nobreak\_medskip\_bigskip \_unvbox0
1386   }%
1387 }
1388 \_skip\.botins=\_zskip % no space added when a topinsert is present
1389 \_count\.botins=1000 % magnification factor (1 to 1)
1390 \_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.

```

1403 \_def\.putImage #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1404   \_edef\.fullvref{\_gentovref{#1}}%
1405   \_edef\.fullvrefm{\_ea\_renumvref\.fullvref\_relax}%
1406   \_ea\_newaction\_ea{\_fullvrefm}{\_doImage{#2}[#4] (#6){#7}}%
1407 }
1408 \_def\.doImage #1[#2] (#3)#4{% {Title}[label] (params){image-file.pdf}
1409   \_botinsert
1410   \_botTitle{#1}[#2]%
1411   \_kern3pt \_nobreak
1412   \_hbox{\picw=\hsize #3\inspic{#4}}%
1413   \_endbot
1414 }
1415 \_def\.botTitle#1[#2]{\_hbox{\_captionfont
```

```

1416 \_ifx^#2^\_else \_botDest{#1}[#2]\_fi
1417 \_rlap{\Grey \_vrule height1.2em depth.5em width\_hsize}\White\_kern12pt #1}%
1418 }
1419 \_picdir={images/}
1420 \_def\_botDest#1[#2]{\_label[#2]\_wlabel{#1}}
1421
1422 \_nspublic \_putImage ;

```

`\putArticle` $\langle chapter \rangle : \langle verse \rangle \{ \langle title \rangle \} [\langle label \rangle] (\langle params \rangle)$ inserts an article (an additional text) given in the file `articles-*.tex` signed by `\Article` $[\langle label \rangle]$. The article starts at the page where $\langle chapter \rangle : \langle verse \rangle$ 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 `\botinsert`. The insert material is breakable at its beginig and between each two-column boxes created by the `\balancecolumns` macro.

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

opbible.opm

```

1446 \_newcount \_articlenum
1447 \_def \_putArticle #1 #2#3[#4]#5(#6){% chap:verse {Title} [number] (params)
1448 \_edef \_fullvref{ \_gentovref{#1} }%
1449 \_edef \_fullvrefm{ \_ea \_renumvref \_fullvref \_relax }%
1450 \_ea \_newaction \_ea{ \_fullvrefm }{ \_doArticle{#2}[#4] (#6) }%
1451 }
1452 \_nspublic \_putArticle ;

```

The `\doArticle` $\{ \langle Title \rangle \} [\langle label \rangle] (\langle params \rangle)$ inserts the article to one or more pages by the pair `\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`.

opbible.opm

```

1469 \_def \_doArticle#1[#2] (#3){% {Title} [number] (params)
1470 \_incr \_articlenum
1471 \_botinsert
1472 \_def \_botDest##1[##2]{ \_trykadedest{ a : \_currbook / ##2 } }
1473 \_parindent=12pt \_iindent=\_parindent
1474 \_setbox0=\_vbox{ \_hsize=.458 \_hsize \_emergencystretch=1em
1475 \_hbadness=6000 \_baselineskip=\_dimexpr \_baselineskip plus1pt
1476 \_def \_Article[##1]{ \_endinput }
1477 \_penalty0
1478 \_long \_def \_searcharticle##1 \_Article[##2] { }
1479 \_ea \_searcharticle \_input \_articlefile \_relax }
1480 \_splittopskip=12pt
1481 \_setbox1=\_vsplit0 toOpt % adding \splittopskip
1482 \_tmpdim=\_vsize \_advance \_tmpdim by -24pt % \_botTitle height plus above/below skips
1483 \_ifdim 2 \_tmpdim > \_ht0 \_tmpnum=1
1484 \_else
1485 \_tmpnum=\_roundexpr{ \_bp{ \_ht0 } / \_bp{ 1.333 \_vsize } + 0.999 } % number of 2/3 pages
1486 \_fi
1487 \_multiply \_tmpnum by 2 % number of columns
1488 \_edef \_Ncols{ \_the \_tmpnum }
1489 \_dimen0=\_expr{ 1 / \_Ncols } \_ht0 \_setbox6=\_box0 % height of each two-columns part
1490 \_setbox0=\_vbox{ \_balancecolumns }
1491 \_tmpdim=\_ht0 \_advance \_tmpdim by 1.2 \_baselineskip
1492 \_setbox0=\_vbox{ \_unvbox0 \_global \_setbox2=\_lastbox }
1493 \_setbox0=\_hbox{ \_unhbox2
1494 \_for num 1.. \_Ncols \_do { \_unskip \_global \_setbox1##1=\_lastbox } }
1495 \_for numstep -2: \_Ncols..1 \_do {
1496 \_hrule height0pt \_kern5pt \_nobreak \_vfill
1497 \_ifnum \_Ncols=##1 \_botTitle{#1}[#2] \_else \_botTitle{ } [ ] \_fi

```

```

1498         \_kern3pt \_nobreak
1499         \_hbox to\_hsize{%
1500             \_rlap{\LightGrey \_vrule height\_tmpdim depth6pt width\_hsize}%
1501             \_kern\_parindent
1502             \_box1##1\_hss\_box1\_the\_numexpr##1-1
1503             \_kern\_parindent
1504         }
1505         \_break
1506     }
1507 \_endbot
1508 }
1509 \_def\_roundexpr#1{\_ea\_roundexprA\_expanded{\_expr{#1}}\_relax}
1510 \_def\_roundexprA#1.#2\_relax{\_ifnum#1=0 0\_else #1\_fi}

```

14 Inserting images over two pages

We can insert an image at the bottom of the page which spans from even to odd page. The macro `\insertSpanImage{<Title>} [<label>] (<params>) {<image file>}` does it. The image is placed at the bottom of the pages using following rule: if the `\insertSpanImage` occurs at the current page c then

- if c is even and the image height fits to the current page then the image is inserted to pages $c, c + 1$,
- if c is even and the image height doesn't fit to the current page then the image is inserted to pages $c + 2, c + 3$,
- if c is odd then the image is inserted to pages $c + 1, c + 2$.

The macro `\insertSpanImage` saves the image in the box `\spanpicbox`. The `\picwidth` of the image is calculated as $2 * (_hsize * (_inner_margin))$. I.e. when we put the box to the page firstly then only the left half of its size is printed.

Next, `\insertSpanImage` checks if the current page is even. If it is true and if there is sufficient space `\pagegoal - \pagetotal` at the current page, the image is inserted to the current page using the `\startinsertSpanImage` which runs `\insertBot` in fact. The second part of the image is printed because `\endoutput` (processed at the end of the output routine where first part of the image is inserted) runs `\addpicbox`. The `\addpicbox` runs second `\insertBot` which is printed on the next page.

If the current page is odd, then `\insertSpanImage` doesn't run `\startinsertSpanImage` immediately, but `\endoutput` inserts first part of the image using `\inspicbox` which is equal to `\inspicboxafter` in this case. It processes `\startinsertSpanImage` which inserts the first part of the image on the next page (even) page.

If the current page is even but the image cannot fit to the current page then the delay using `\endoutput` is activated too. But the `\inspicboxafter` checks that the current page is even and it does nothing in this case. Next page is odd, so `\inspicboxafter` invoked by next `\endinput` inserts the first part of the image which will be printed on the next (even) page.

opbible.opm

```

1556 \_newbox \spanpicbox
1557
1558 \_def\_insertSpanImage #1#2[#3]#4(#5)#6{%
1559     \_checkpicbox
1560     \_par \_penalty0
1561     \_tmpdim=\_pagewidth
1562     \_advance\_tmpdim by-\_hoffset
1563     \_global\_setbox\spanpicbox=\_hbox{\_picwidth=2\_tmpdim \_inspic{#6}}
1564     \_gdef\_startinsertSpanImage {\_insertBot {#1}[#3] (#5){\_copy\spanpicbox \_kern-1.2ex}}
1565     \_doinsertSpanImage
1566 }
1567 \_def\_doinsertSpanImage{%
1568     \_ifodd\_pageno
1569         \_glet\_inspicbox=\_inspicboxafter
1570     \_else
1571         \_ifdim \_dimexpr \_pagegoal-\_pagetotal > \_dimexpr \_ht\spanpicbox+2em \_relax
1572         \_startinsertSpanImage
1573     \_else
1574         \_glet\_inspicbox=\_inspicboxafter
1575     \_fi
1576 \_fi
1577 }

```

```

1578 \_let\inspicbox=\_useit
1579 \_def\inspicboxafter #1{%
1580   \_ifodd\_pageno
1581     \_startinsertSpanImage
1582     \_glet\inspicbox=\_useit
1583   \_fi
1584 }
1585 \_def \_endoutput{%
1586   \_ifvoid\spanpicbox\_else \_addpicbox\_fi
1587   \_advancepageno
1588   {\_globaldefs=1 \_the\_nextpages \_nextpages={}}%
1589   \_ifnum\_outputpenalty>-20000 \_else\_dosupereject\_fi
1590 }
1591 \_def\addpicbox{\_inspicbox{\_insertBot{ }[] (){\_moveleft\_pagewidth\_box\spanpicbox\_kern-1.2ex}}
1592 }
1593 \_def\checkpicbox{%
1594   \_ifvoid\spanpicbox\_else \_errmessage{Two span Image/Text at single place not allowed}\_fi
1595 }

```

`\insertSpanText{<Title>} [label] (<params>) {<text>}` does the same as `\insertSpanImage`, but the *text* is inserted instead the image. The `\hsize` is locally set to the desired width of the text when *text* is processed in a `\vbox`, i.e. to $2*(\hsize + \textit{inner_margin})$.

```

1605 \_long\_def\insertSpanText #1#2[#3]#4(#5)#6{%
1606   \_checkpicbox
1607   \_par \_penalty0
1608   \_tmpdim=\_pagewidth
1609   \_advance\_tmpdim by-\_hoffset
1610   \_setbox0=\_hbox to2\_tmpdim{\_hss\_vbox{\_hsize=2\_tmpdim
1611     \_leftskip=0pt \_rightskip=0pt \_relax \_kern3pt #6}\_hss}
1612   \_global\_setbox\spanpicbox=
1613     \_hbox{\_rlap{\_White \_vrule width\_wd0 height\_ht0 depth\_dp0}\_box0}
1614   \_global\_ht\spanpicbox=\_dimexpr\_ht\spanpicbox-3pt\_relax
1615   \_gdef\startinsertSpanImage {\_insertBot {#1} [#3] (#5){\_copy\spanpicbox \_kern-1.2ex}}
1616   \_doinsertSpanImage
1617 }
1618 \_nspublic \insertSpanImage \insertSpanText ;

```

`\putSpanImage <chapter>:<verse> {<title>} [label] (<params>) {<img-file>}` runs `\insertSpanImage` at the page where the beginning of the verse given by *chapter*:<verse> exists. We register a new action by `\newaction{<full-vref>}{_doSpanImage{<title>} [label] (<params>) {<img-file>}}`.

`\putSpanText <chapter>:<verse> {<title>} [label] (<params>) {<text>}` runs `\insertSpanText` at the page where the beginning of the verse given by *chapter*:<verse> exists. The *text* is saved to `\spant!\the\spantxtnum` and only the name of this macro is registered by the `_newaction`.

Note that the image/text itself is inserted at the current page *c* and *c* + 1 or at *c* + 1, *c* + 2 or at *c* + 2, *c* + 3.

```

1634 \_newcount\spantextnum
1635 \_def\putSpanImage #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1636   \_edef\fullvref{\_gentovref{#1}}%
1637   \_edef\fullvrefm{\_ea\_renumvref\fullvref\_relax}%
1638   \_ea\_newaction\_ea{\_fullvrefm}{\_insertSpanImage{#2} [#4] (#6) {#7}}%
1639 }
1640 \_long\_def\putSpanText #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1641   \_edef\fullvref{\_gentovref{#1}}%
1642   \_edef\fullvrefm{\_ea\_renumvref\fullvref\_relax}%
1643   \_incr\spantextnum
1644   \_global\_sdef{spant!\the\spantextnum}{#7}%
1645   \_ea\putSpanTextA
1646   \_expanded{\_fullvrefm}\_ea\_csname spant!\the\spantextnum\_endcsname {#2} [#4] (#6)%
1647 }
1648 \_def\putSpanTextA #1#2#3[#4] (#5){\_newaction{#1}{\_insertSpanText{#3} [#4] (#5) {#2}}}
1649 }
1650 \_nspublic \putSpanImage \putSpanText ;

```


15 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>}}`.

```

1662 \_def\putCite #1 #2{% chap:verse {text}
1663   \_edef\fullvref{\gentovref{#1}}%
1664   \_edef\fullvrefm{\_ea\renumvref\fullvref\relax}%
1665   \_ea\.newaction\_ea{\fullvrefm}{\dotopCite{#2}}%
1666 }
1667 \_nspublic \putCite ;

```

opbible.opm

`\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 save 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.

```

1679 \_newcount\citenum
1680 \_def\dotopCite #1{%
1681   \topinsertnopar
1682   \_typosize[12/16]\_bi
1683   \_incr\citenum
1684   \_ifodd \_trycs{ct!\_the\citenum}{0}\_relax
1685     \_leftskip=.3\_hsize plus1fil \_parfillskip=0pt
1686     \_noindent
1687     \rlap{\_hskip\_hsize \_kern-\_leftskip \_copy\lqqbox}\_hfill
1688   \_else
1689     \_let\quotedby=\_quotedbyright
1690     \_rightskip=.3\_hsize plus 1fil
1691     \_noindent \_llap{\_copy\lqqbox}%
1692   \_fi
1693   {\_printCite{#1}\_unskip}\_par
1694   \_ewref\_sxdef{ct!\_the\citenum}{\_string\mypage}}%
1695 % \_vskip-.3\_baselineskip
1696 \_endinsert
1697 }
1698 \_def\printCite#1{\_pdfliteral{2 Tr .15 w .9 g}#1\_pdfliteral{0 Tr 0 w 0 g}}
1699 \_def\printCite#1{{\Grey#1}}
1700
1701 \_def\_topinsertnopar{\_umidfalse \_upagefalse \_begingroup\_setbox0=\_vbox\_bgroup\_resetattrs}

```

opbible.opm

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.

```

1711 \_newbox\lqqbox
1712 \_newbox\rqqbox
1713 \_setbox\lqqbox=\_hbox{\_lower3pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed,}}
1714 \_setbox\rqqbox=\_hbox{\_kern2pt\_lower38pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed"}}
1715 \_ht\lqqbox=0pt \_dp\lqqbox=0pt
1716 \_ht\rqqbox=0pt \_dp\rqqbox=0pt
1717 \_addto\enquotes{\_setbox0=\_box\lqqbox \_setbox\lqqbox=\_box\rqqbox \_setbox\rqqbox=\_box0 }
1718
1719 \_def\quotedby{\_par}
1720 \_def\quotedbyright#1{%
1721   \_unskip\_nobreak\_hfill\_penalty0\_hskip2em
1722   \_null\_nobreak\_hskip\_iindent\_hbox{#1}}

```

opbible.opm

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>`.


```

1736 \_def\Cite #1#2{\_sdef{c!\_the\articlenum!#1}{#2}}
1737 \_def\insertCite #1#2{\_def\citelabel{#1}%
1738 \_ifx\_left#2\insertCiteleft
1739 \_else \_ifx#2\_right\insertCiteright\_else
1740 \_errmessage{\_noexpand\insertCite#1: \_noexpand\left or \_noexpand\right expected}%
1741 \_fi\_fi
1742 }
1743 \_def\insertCiteleft {%
1744 \_ifnum\citepg=1
1745 \_printwarn{\_noexpand\insertCite\citelabel: \_noexpand\swapCites activated}\_fi
1746 \_ifodd \_numexpr\_trycs{cp!\_the\articlenum!\_citelabel}{0}+\_citepg\_relax
1747 \_else \_insertCitelr \_left \_fi
1748 }
1749 \_def\insertCiteright{%
1750 \_ifodd \_numexpr\_trycs{cp!\_the\articlenum!\_citelabel}{0}+\_citepg\_relax
1751 \_insertCitelr \_right \_fi
1752 }
1753 \_def\insertCitelr#1{\_unskip\_vadjust{\_vbox{%
1754 \_ewref\_sxdef{{cp!\_the\articlenum!\_citelabel}{\_string\mypage}}%
1755 \_vskip6pt
1756 \_advance\_hsize by\_parindent
1757 \_typosize[12/16]\_bi\Grey
1758 \_ifx#1\_left
1759 \_def\quotedby{\_par\_hfill}
1760 \_rightskip=\_parindent plus1fil \_leftskip=0pt
1761 \_setbox0\_vbox{%
1762 \_medskip \_noindent
1763 \_llap{\_copy\lqqbox}\_ignorespaces
1764 \_printCite{\_cs{c!\_the\articlenum!\_citelabel}}\_medskip}%
1765 \_hbox{\_kern-\_parindent\_rlap{White
1766 \_vrule height\_ht0 width\_hsize}\_box0}%
1767 \_else
1768 \_leftskip=\_parindent plus1fil
1769 \_parfillskip=0pt
1770 \_setbox0\_vbox{%
1771 \_medskip \_noindent
1772 \_rlap{\_hskip\_hsize\_kern-\_parindent\_copy\rqqbox}\_hfill
1773 \_ignorespaces \_printCite{\_cs{c!\_the\articlenum!\_citelabel}}\_medskip}%
1774 \_rlap{\_rlap{White \_vrule height\_ht0 width\_hsize}\_box0}%
1775 \_fi
1776 \_vskip6pt
1777 }}}
1778 \_def\swapCites{\_def\citepg{1}}
1779 \_def\citepg{0}
1780
1781 \_nspublic \Cite \insertCite ;

```

Insertions into the intro text

```

1789 %% TBN page 236
1790
1791 \_newcount\shapenum
1792 \_newdimen\ii \_newdimen\w
1793 \_def\oblom #1 od #2 odsadit #3 {\_par \.ii=#1 \.w=\_hsize
1794 \_ifdim\ii>\_zo \_advance\w by-\_ii
1795 \_else \_advance\w by\_ii \.ii=\_zo \_fi
1796 \.shapenum=1 \_tmpnum=0 \_def\shapelist{}
1797 \_loop \_ifnum\shapenum<#2 \_edef\shapelist{\.shapelist\_zo\_hsize}%
1798 \_advance\shapenum by1 \_repeat
1799 \_loop \_edef\shapelist{\.shapelist\ii.w}%
1800 \_advance\_tmpnum by1 \_ifnum\_tmpnum<#3 \_repeat
1801 \_advance\shapenum by#3 \_edef\shapelist{\.shapelist\_zo\_hsize}
1802 \.doshape}
1803 \_def\doshape{\_parshape \shapenum \shapelist}
1804 \_newcount\globpar
1805 \_ifx\_partokenset \_undefined \_def\partoken{\_par} \_else \_def\partoken{\_par} \_fi
1806 \_def\doshape{\_global\globpar=0 \_ea\_def\partoken{\_ifhmode\shapepar\_fi}}
1807 \_def\shapepar{\_prevgraf=\_globpar \_parshape\shapenum\shapelist
1808 \_endgraf \_global\globpar=\_prevgraf
1809 \_ifnum \_prevgraf>\shapenum \_ea\_let\partoken=\_endgraf \_fi

```

```

1810 }
1811
1812 \_def\Citehereleft #1 (#2) #3{{
1813   \_par
1814     \_def\quotedby{\_par\_hfill}
1815     \_rightskip=\_parindent plus1fil \_leftskip=0pt
1816     \_setbox0\_vbox{{%
1817       \_typosize[12/16]\_bi\Grey
1818       \_hsize=.5\_hsize
1819       \_medskip \_noindent
1820       \_llap{\_copy\lqqbox}\_ignorespaces
1821       \_printCite{#3}\_medskip}}%
1822   \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1823   \_xdef\.lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1824   \_nointerlineskip\_vbox toOpt{\_kern#1\_baselineskip #2
1825     \_hbox{\_rlap{White
1826       \_kern-3mm\_vrule height\_ht0 width.5\_hsize}\_box0}}%
1827   \_vss}}
1828   \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1829   \_oblom {.5\_tmpdim} od #1 odsadit {\.lines}
1830 }
1831 \_def\Citehereright #1 (#2) #3{{
1832   \_par
1833     \_def\quotedby{\_par\_parfillskip=0pt \_hfill}
1834     \_leftskip=\_parindent plus1fill \_rightskip=0pt
1835     \_setbox0\_vbox{{%
1836       \_typosize[12/16]\_bi\Grey
1837       \_hsize=.5\_hsize
1838       \_vskip\_medskipamount \_rlap{\_kern\_hsize\_copy\lqqbox}\_vskip-\_medskipamount
1839       \_printCite{\_noindent\_ignorespaces#3}\_medskip}}%
1840   \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1841   \_xdef\.lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1842   \_nointerlineskip\_vbox toOpt{\_kern#1\_baselineskip #2
1843     \_hbox to\_hsize{\_hss
1844       \_llap{White \_vrule height\_ht0 width.5\_hsize \_kern-3mm}%
1845       \_llap{\_box0}}
1846   \_vss}}
1847   \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1848   \_oblom {-.5\_tmpdim} od #1 odsadit {\.lines}
1849 }
1850
1851 \_def\Citehere{\_par \_ifodd\_pageno \_ea\Citehereright \_else \_ea\Citehereleft \_fi}
1852
1853 \_nspublic \Citehere ;

```

`\insertBot` $\langle title \rangle$ [$\langle label \rangle$] ($\langle params \rangle$) $\langle data \rangle$ inserts a material from $\langle data \rangle$ to the bottom of the current page or next page if it is unable to fit to the current one. The material is titled by $\langle title \rangle$ and it can be referred by $\langle label \rangle$. The $\langle params \rangle$ can include a special setting used locally for the printing of this material.

`\putBot` $\langle chapter \rangle$: $\langle verse \rangle$ $\langle title \rangle$ [$\langle label \rangle$] ($\langle params \rangle$) $\langle data \rangle$ behaves like `\insertBot`, but the result is printed to the bottom of the page where the verse $\langle chapter \rangle$: $\langle verse \rangle$ is, or to the next page if the material is unable to fit to the current one.

opbible.opm

```

1869 \_def\insertBot #1#2[#3]#4(#5)#6{% {Title} [label] (params) {data}
1870   \_botinsert
1871     \_leftskip=0pt \_rightskip=0pt \_relax
1872     \_botTitle{#1}[#3]%
1873     \_kern3pt \_nobreak
1874     \_vbox{\_picwidth=\_hsize #5 #6}%
1875   \_endbot
1876 }
1877 \_def\putBot #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1878   \_edef\.fullvref{\_gentovref{#1}}%
1879   \_edef\.fullvrefm{\_ea\renumvref\.fullvref\_relax}%
1880   \_ea\newaction\_ea{\.fullvrefm}\_insertBot{#2}[#4](#6){#7}}%
1881 }
1882 \_nspublic \insertBot \putBot ;

```

`\.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`.

opbible.opm

```
1891 \_def\.printintro{%
1892   \.begblock
1893     \_dest[i:\.currbook/]
1894     \_chaptit{\_mtext{intro}}%
1895     \_input{\introfile}
1896   \.endblock
1897 }
```

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.

opbible.opm

```
1905 \_newcount\.blocklevel % nesting level of blocks
1906 \_def\.begblock{\_par\_bgroup
1907   \_advance\.blocklevel by1 \_advance\_leftskip by\_iindent \_rightskip=\_leftskip
1908   \_medskip
1909   \_pdfsavepos \_ea\_wref\_ea\.Xblock\_ea{\_ea{\_the\.blocklevel}B{\_the\_pdflastypos}}
1910   \_nbreak \_medskip
1911 }
1912 \_def\.endblock{\_par\_nbreak\_medskip
1913   \_pdfsavepos \_ea\_wref\_ea\.Xblock\_ea{\_ea{\_the\.blocklevel}E{\_the\_pdflastypos}}
1914   \_medskip \_egroup
1915 }
1916 \_refdecl{%
1917   \_def\.Xblock#1#2#3{\_ifnum#1=1 \_edef\.tmp{frm:\_ea\_ignoresecond\_currpage}^^J
1918     \_unless\_ifcsize \_tmp \_endcsize \_sxddef{\_tmp}{\_fi^^J
1919     \_sxddef{\_tmp}{\_cs{\_tmp}#2#3}\_fi}
1920 }
1921 \_newdimen\.frtop \_newdimen\.frbottom % positions of top and bottom text on the pages
1922 \_def\.frcolor{.93 g } % light grey -- color of blocks.
1923 \_pgbackground={%
1924   \_slet{\_opb\_tmp}{frm:\_the\_gpageno}
1925   \_ifx\_tmp\_undefined \_def\.tmp{\_fi}
1926   \_frtop=\_dimexpr \_pdfpageheight-\_voffset+\_smallskipamount\_relax
1927   \_frbottom=\_dimexpr \_pdfpageheight-\_voffset-\_vsize-\_medskipamount\_relax
1928   \_ifx\_frnext y \_edef\.tmp{B{\_number\.frtop}\_tmp}\_global\_let \_frnext n\_fi
1929   \_ea\.printframes \_tmp B{0}E{\_number\.frbottom}
1930   \_ifx\.frameslist\_empty \_else
1931     \_pdfliteral{q \_frcolor 1 0 0 1 0 \_bp{-\_pdfpageheight} cm \_frameslist Q}\_fi
1932 }
1933 \_def\.printframes B#1#2E#3{\_ifnum#1=0 \_else
1934   \_printframe {\_hoffset}{#3sp}{\_xhsize}{\_ifnum#1=-1 \_number\.frtop\_else#1\_fi sp-#3sp}
1935   \_ifx^#2^\_else \_global\_let \_frnext=y \_let\.printframes=\_relax \_fi
1936   \_ea\.printframes\_fi}
1937 }
1938 \_def\.frameslist{}
1939 \_def\.printframe #1#2#3#4{\_edef\.frameslist{\_frameslist
1940   \_bp{#1} \_bp{#2} \_bp{#3} \_bp{#4} re f }%
1941 }
```

Insertions objects over pictures (maps)

`\putstext` $\langle x-pos \rangle \langle y-pos \rangle \{ \langle text \rangle \}$ behaves like `\puttext` from OpTeX, but moreover, it inserts a "white shadow" as a background of the text. It can be used as text printed over a pictures (maps etc.).

`\shadowedtext` $\{ \langle text \rangle \}$ creates an `\hbox` $\{ \langle text \rangle \}$ with "white shadow" as background.

`\shadowparameter` is a number of "transparency amount" used for "white shadows". User can re-define it but it must be done before first usage of `\putstext` or `\shadowedtext` and it is used for whole document.

opbible.opm

```
1962 \_def\.putstext{\_ea\_ea\_ea\.putstextA\_scantwodimens}
1963 \_def\.putstextA#1#2#3{%
1964   \_setbox0=\_hbox{\_shadowedtext{#3}}%
1965   \_dimen1=#1sp \_dimen2=#2sp \_puttextB
1966 }
1967 \_def\.shadowedtext#1{%
1968   \_insertwhiteshadowresources
```

```

1969 \_setbox0=\_hbox{#1}%
1970 \_hbox{\_tmpdim=\_ht0 \_advance\_tmpdim by\_dp0
1971 \_lower\_dp0\_hbox{%
1972 \_pdfliteral{q /trans gs 1 g
1973 \_fornum 1..10\_do{\_oval{\_bp{\_wd0}}{\_bp{\_tmpdim}}{2+##1/2} f } Q}}%
1974 \_box0}%
1975 }
1976 \_def\insertwhiteshadowresources{%
1977 \_addextgstate{trans}{<</ca \shadowparameter>>}%
1978 \_glet\insertwhiteshadowresources=\_relax
1979 }
1980 \def\shadowparameter{.1} % default value of "transparency"
1981
1982 \_nspublic \putstext \shadowedtext ;

```

`\c[(init-rot)/(step)]{(text)}` prints the *(text)* around a curve. Each letter or space from *(text)* is processed individually. The first letter is rotated by *(init)* degrees. Next letters are printed after *(step)* transformation is applied.

opbible.opm

```

1991 \_def\c[#1/#2]#3{% text podel krivky: \c[init-rotace/repetice]{text}
1992 \_pdfsave\_pdfrotate{#1}\_rlap{\_let\printwarn=\_ignoreit
1993 \edef\tmpb{#3}\_replstring\_.tmpb{ }{ } \_def\tpa{#2}%
1994 \_ea\_foreach\_.tmpb\_do{##1\_.tpa}}\_pdfrestore \_kern10mm
1995 }
1996 \_let\c=\_undefined
1997 \_nspublic \c ;

```

`\town <dimen> <dimen>` puts a circle with given `\townparams` to the given place *(dimen)* *(dimen)*. It works like `\putstext <dimen> <dimen> {<circle>}`.

opbible.opm

```

2005 \_def\townparams{ % default parameters of the circle:
2006 \_hhkern=.8pt % diameter of the disc
2007 \_lwidth=.5pt % tickness of the outline
2008 \_fcolor=\Red % color of the inner disc
2009 \_lcolor=\Black % color of the outline
2010 }
2011 \_def\town {\_ea\_ea\_ea\_.townA\_scantwodimens}
2012 \_def\townA #1#2{\_setbox0=\_hbox{\_incircle[\_hhkern=0pt \_vbkern=0pt \townparams]{} }%
2013 \_dimen1=#1sp \_dimen2=#2sp \_putstextB
2014 }
2015 \_nspublic \town ;

```

16 Chiasm

The pair `\begChiasm... \endChiasm` defines chiasm environemnt. It behaves like `\begitems... \enditems`, but you can use given number of * which denotes the indentation level. The letters A, B, C, etc. will be prefixed automatically and when you are in the backward phase then C', B', A' are prefixed. You can try:

```

\begChiasm
* Předkové a rané zkušenosti (\<11:10-12:9>)
** Rané kontakty s ostatními národy (\<12:10-14:24>)
*** Smlouva s Bohem (\<15:1-17:27>)
** Pozdní kontakty s ostatními národy (\<18:1-21:34>)
* Potomci a smrt (\<22:1-25:18>)
\endChiasm

```

opbible.opm

```

2038 \_def\easylist{\_ade*{\_countlist}}
2039 \_def\.aast{\_countlist}
2040 \_def\countlist{\_tmpnum=1 \_countlistA}
2041 \_def\countlistA{\_futurelet\_.next\_countlistB}
2042 \_def\countlistB{\_ifx\_.next\_.aast \_ea\_countlistC\_else \_ea\_countlistD \_fi}
2043 \_def\countlistC#1{\_incr\_tmpnum \_countlistA}
2044 \_def\countlistD{%
2045 \_ifnum\_tmpnum>\_ilevel \_fornum \_ilevel..\_tmpnum-1 \_do{\_begitems\_.easylist}\_else
2046 \_ifnum\_tmpnum<\_ilevel \_fornum \_tmpnum..\_ilevel-1 \_do{\_enditems}\_fi\_fi

```

```

2047 \_startitem}
2048
2049 \_def\qq#1{\_bf#1\_trycs{Level:\_the\_ilevel}{}}\_space\_aftergroup\qqA}
2050 \_def\qqA{\_sdef{Level:\_the\_ilevel}{\_rlap'}}}
2051 \_def\ChiasmNumbering{\_ea\qq \_Uchar \_numexpr `A-1+\_ilevel\_relax\_space} % A, B, C, D, etc.
2052 \_sdef{item:q}{}%for chiasms with no leading alphabet letters
2053 \_sdef{item:Q}{\_ChiasmNumbering}
2054 \_def\beginChiasm{\_begitems \_easylist \_style Q \_let\_defaultitem=\_printitem}
2055 \_def\endChiasm{\_fornum 1..\_ilevel \_do{\_enditems}}
2056
2057 \_nspublic \beginChiasm \endChiasm ;

```

17 Outline

The `\Outline` starts two column format in the introduction text. Nested lists are printed into the first column and comments declared by `\rightnote{comment}` are printed to the right column.

opbible.opm

```

2069 \_newdimen\colsep
2070 \colsep=10pt
2071
2072 \_def\Outline{
2073 \_medskip
2074 % \filbreak
2075 \chaptit{\_mtext{outline}}%
2076 \_everylist={\_ifcase\_ilevel \_or \_style I \_or \_style A \_or \_style n \_fi}
2077 \_sdef{item:A}{\_strut\_uppercase\_ea{\_athe\_itemnum}. }
2078 \_sdef{item:I}{\_strut\_uppercase\_ea{\_romannumeral\_itemnum}. }
2079 \_hsize=.5\_hsize \_advance\_hsize by-\colsep
2080 \_emergencystretch=40pt
2081 \_leftskip=0pt \_rightskip=0pt
2082 }
2083 \_def\rightnote#1{\_par
2084 \_setbox0=\_hbox{\_kern\_hsize \_kern\colsep
2085 \_vtop{\_leftskip=0pt \_kern0pt\_noindent\_strut\_it#1}}
2086 \_ht0=0pt \_dp0=0pt \_box0 \_nointerlineskip
2087 }
2088 \_nspublic \Outline \rightnote ;

```

18 Timelines

- `\timeline{num}` sets the total number of years (or other units) in time-line.
- `\timelinewidth{dimen}` sets the width of time-line.
- `\l` is shortcut for `\baselineskip` (an be used in `\vskip` parameter).

opbible.opm

```

2101 \_def\l{\_baselineskip}
2102 \_newcount\timeline \_timeline=100 % default
2103 \_newdimen\tlwidth \_tlwidth=10cm % default
2104 \_def\timelinewidth{\_afterassignment\_timelinewidthA\_tlwidth}
2105 \_def\timelinewidthA{\_par\_hbox to\_tlwidth{}}
2106
2107 \_let\l=\_undefined
2108 \_nspublic \l \timeline \timelinewidth ;

```

All objects used for creating time-line are defined by `\puttext`, i.e. they don't shift the current typesetting point.

`\arrowtext {from}..{to} ((settings)) {text}` creates a horizontal line with arrows. Its width and its position is given by `{from}..{to}` time units. The `{settings}` can include font selector, color settings of something similar for `{text}`. The `{text}` is placed to the center of the line.

opbible.opm

```

2121 \_def\arrowtext #1.#2(#3)#4{%
2122 \_puttext \_pos{#1}0pt
2123 {\_lower.745ex\_hbox to\_dimexpr\_pos{#2}-\_pos{#1}{#3}\_Larrow{ #4 }\_Rarrow}}
2124 }
2125 \_def\larrow{\_leftarrow$\_kern-.8em\_leaders\_vrule height.65ex depth-.42ex\_hfil}
2126 \_def\rarrow{\_leaders\_vrule height.65ex depth-.42ex\_hfil\_kern-.8em$\_rightarrow$}

```

```

2127 \_def\.rule{\_leaders\_vrule height.12ex depth.12ex\_hfil}
2128 \_def\.pos#1{\_expr{#1/\_the\.timeline}\.tlwidth}
2129
2130 \_nspublic \arrowtext ;

```

\tlput *<above/below>* *<where>* *<llap or rlap or nothing>* (*<format of text>*) *{<text>}* puts the *<text>* to the timeline. The *<text>* can include more lines separated by `\cr`. The parameter *<above/below>* is `a` or `b` and means the *<text>* position: above the current point or below it. *<where>* is the position of the text in time units. *<llap or rlap>* is `\llap` or `\rlap` and it means that text is encapsulated to `\llap`, `\rlap`. If nothing is here the text is centered. The *<format of text>* can include the font setting, color setting etc.

opbible.opm

```

2143 \_def\.tlput #1 #2 #3(#4)#5{%
2144   \_let\.Lhss=\_hss \_let\.Rhss=\_hss
2145   \_ifx#3\_rlap\_relax \_let\.Lhss=\_relax \_let\.Rhss=\_hss \_fi
2146   \_ifx#3\_llap\_relax \_let\.Lhss=\_hss \_let\.Rhss=\_relax \_fi
2147   \_puttext \_pos{#2}Opt {\_hbox toOpt{\_Lhss #4\.tltext#1{#5}\_Rhss}}
2148 }
2149 \_def\.tltext#1#2{\_ifx#1a\_vbox\_else
2150   \_vtop\_fi{\_kern0pt\_halign{\_Lhss#\_Rhss\_cr\_strut#2\_crr}}}%
2151 }
2152 \_nspublic \tlput ;

```

\tline*<from>*..*<to>* prints the line. Its length and position is given by *<from>*..*<to>* time units.

\tlines*{<data/separated/by/>}* creates a list of short vertical lines. Each line is represented by one |. The distance between lines (in time units) are given in the parameter.

opbible.opm

```

2162 \_def\.tline #1..#2 {%
2163   \_puttext \_pos{#1}Opt {\_hbox to \_dimexpr\_pos{#2}-\_pos{#1}{\_rule}}
2164 }
2165 \_def\.tlines#1{\_puttext OptOpt{\_hbox{\_foreach #1|\_do##1|{\_vrule\_hskip\_pos{0##1}}}}
2166 \_def\.vrule{\_def\.vrule{\_kern-.12ex\_vrule height.7\1 depth.7\1 width.24ex \_kern-.12ex}}
2167
2168 \_nspublic \tline \tlines ;

```

19 Typesetting variants

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

opbible.opm

```

2182 \_def\.normalchapnumbers{
2183   \_margins/2 a4 (25,25,20,20)mm
2184   \_lrmargin=0pt
2185   \_setbox0=\_box\lqqbox \_setbox0=\_box\rqqbox
2186   \_def\.printbeforefirst{%
2187     \_nobreak\_medskip
2188     \_trychapnote
2189     \_hangindent=\_parindent \_hangafter=-2
2190     \_noindent \_llap{\_vbox toOpt
2191       {\_kern-8pt\_hbox{\_setfontsize{at23pt}\_bf\Red\_the\.chapnum\_kern5pt}\_vss}}%
2192   }
2193 }
2194 \_nspublic \normalchapnumbers ;

```

20 Checking syntax

opbible.opm

```

2202 \_def\.checksyntax#1 {%
2203   \_let\processbooks=\_relax
2204   \_ifx\_relax#1\_relax \_else
2205     \_begingroup
2206       \_the\.syntaxmacros
2207       \_wterm{^^J** checking file: #1 **^^J}
2208       \_input{#1}
2209       \_vfil\_break
2210     \_endgroup

```

```

2211 \_ea\checksyntax \_fi
2212 }
2213
2214 \_newtoks\syntaxmacros
2215 {\_catcode`<=13
2216 \_global\syntaxmacros={
2217 \_def<#1>{\_bgroup
2218 \_message{checking \_unexpanded{<#1>}}%
2219 \_ifx\_relax#1\_relax \_errmessage{empty link}\.nobref\_else \_afterfi{\.checkbref#1>\.bref#1>}\_fi
2220 \_glet\linkpre=\linkpre \_glet\linkfspec=\linkfspec
2221 \_egroup
2222 }
2223 \_def\checkbref#1#2>{%
2224 \_isinlist{.#1#2}{<}\_iftrue \_errmessage{duplicated \_string<}\.nobref\_else
2225 \_ifx"#1\checkbrefQ #1#2>\_else \.checkbrefD #1#2>\_fi\_fi
2226 }
2227 \_def\checkbrefQ "#1"#2#3>{\.checkbrefD #2#3>}
2228 \_def\checkbrefD #1>{%
2229 \_isinlist{.#1}{ }\_iftrue\checkbrefS#1>\_else\checkbrefN#1>\_fi
2230 }
2231 \_def\checkbrefS #1 #2>{\.checkbrefN#2>}
2232 \_def\checkbrefN #1>{%
2233 \_def\tmpb{#1}
2234 \_ifx\tmpb\_empty \_errmessage{missing link data}\.nobref\_else
2235 \_replstring\tmpb{:}{ }\_replstring\tmpb{-}{ }\_replstring\tmpb{ }{ }%
2236 \_replstring\tmpb{a}{ }\_replstring\tmpb{b}{ }\_replstring\tmpb{c}{ }%
2237 \_setbox0=\hbox{\_tmpnum=0\tmpb\_relax}%
2238 \_ifdim\_wd0>opt \_errmessage{nonnumeric link data}\.nobref\_fi
2239 \_fi
2240 }
2241 \_def\nobref{\_def\.bref##1>{\Red\_string<##1>}}
2242 \_def\currbook{}
2243 \_def\prelinkB{BK}
2244 \_def\prelinkC{BK}
2245 \_def\prelinkV{0}
2246 \_def\nochapbooks{BK}
2247 \_let<=<
2248
2249 \_def\x/#1/{\_def\tmpb{#1}%
2250 \_isinlist\tmpb\x\_iftrue \.badx
2251 \_else \_isinlist\tmp<\_iftrue \.badx
2252 \_else \_isinlist\tmp\enditems\_iftrue \.badx \_else \.x/#1/\_fi\_fi\_fi
2253 }
2254 \_def\badx{\_errmessage{unclosed \_string\x/.../}}
2255
2256 \_def\Article[#1]{ }
2257 \_def\Cite #1 {\_par\_noindent{\_bf Cite: }}
2258 \_def\insertCite #1#2{}
2259
2260 \_def\putArticle #1 #2[#3]#4(#5){ }
2261 \_def\putCite #1:#2 {\_par\_noindent{\_bf Cite: }}
2262 \_def\putBot #1 #2[#3]#4(#5){\_vbox}
2263
2264 \_def\c[#1/#2]#3{#3}
2265
2266 \_long\_ea\_def\_csname Note\_endcsname #1 #2#3%
2267
2268 {\_par \_let\nextww\_undefined \_noindent{\_bf Note #1:} #3\_par}
2269 }}
2270 \_nspublic \checksyntax ;

```

21 Generating templates from templates

The `\filegen{<file-name-template>}{<cr>}<file-content-template>{<cr>}\endfile` saves `<file-name-template>` to `\.filename` and `<file-content-template>` to `\.filecontent`. Then it runs a loop over `\genbooks`. The `\genbooks` macro is defined by `\BookTitle` and user can re-define it.

The `\.btitle{<bmark or amark>}` expands to full title of the given book.


```

2285 \_newwrite\outfile
2286 \_def\filegen #1 {\_par
2287   \_begingroup \_addto\genbooks{ }\_def\filename{#1}%
2288   \_setverb \_endlinechar=^^J \filegenA
2289 }
2290 \_ea\_def \_ea\filegenA \_expanded{#1^^J\_csstring\endfile#2^^J}{%
2291   \_def\filecontent{#1}%
2292   \_ea\_foreach\genbooks \_do ##1 {%
2293     \_bgroup
2294     \_ifx^##1\_else
2295     \_replstring\filename{@@}{##1}%
2296     \_isfile{\filename}\_iftrue \_opwarning{file "\filename" exists already}%
2297     \_else
2298     \_wterm{creating file: \filename}%
2299     \_immediate\_openout\outfile={\filename}%
2300     \_replstring\filecontent{@@}{\btitle{##1}}%
2301     \_replstring\filecontent{@@}{##1}%
2302     \_immediate\_write\outfile{\filecontent}\_immediate\_closeout\outfile
2303     \_fi\_fi
2304     \_egroup
2305   }%
2306   \_endgroup
2307 }
2308 \_def\btitle#1{\_ifcsname fb!#1\_endcsname \_trycs{btitle!\_cs{fb!#1}}{#1}%
2309   \_else \_trycs{btitle!#1}{#1}\_fi
2310 }
2311 \_nspublic \filegen ;

```

22 Other macros

The temporary macros are here. Maybe, they will be (more conceptually) rewritten.

```

2321
2322 \_def\quotationmarks#1#2{%
2323   \_cnvtext{"}{\_doquotationmark}%
2324   \_def\doquotationmark {\_futurelet\_next\doquotationmarkA}%
2325   \_def\doquotationmarkA {%
2326     \_let\doquotationmarkB=#1\_relax
2327     \_ea\_ifx\_space\_next \_let\doquotationmarkB=#2\_fi
2328     \_ifx\_space\_next \_let\doquotationmarkB=#2\_fi
2329     \_ifx\_endgraf\_next \_let\doquotationmarkB=#2\_fi
2330     \_ifx\_empty\_next \_let\doquotationmarkB=#2\_fi
2331     \_ifx.\_next \_let\doquotationmarkB=#2\_fi
2332     \_ifx,\_next \_let\doquotationmarkB=#2\_fi
2333     \doquotationmarkB}%
2334 }
2335 \_nspublic \quotationmarks ;
2336
2337 \_def\chaptit#1{\_line{\_hss\chapfont\Red#1\_hss}
2338   \_nobreak
2339 }
2340 \_def\schaptit#1{\_bigskip\chaptit{#1}\_nobreak\_medskip}
2341
2342 \_def\subtit#1{\_par
2343   \_ifnum\currversenum=1 \_else \_medskip\_fi
2344   \_line{\_indent\subtitfont #1\_hss}\_nobreak
2345   \_ifnum\currversenum=1 \_vskip-\_medskipamount\_fi
2346   \_smallskip
2347 }
2348 \_def\subtitfont {\Red\_it}
2349
2350 \_nspublic \chaptit \schaptit \subtit ;
2351
2352 \_sdef{\_mt:intro:en}{Introduction} \_sdef{\_mt:outline:en}{Outline}
2353 \_sdef{\_mt:intro:cs}{Úvod} \_sdef{\_mt:outline:cs}{Osnova}
2354
2355 \_def\dopsat{\Red !!! DOPSAT !!! }
2356

```

```

2357 \_def\.bibleinput#1 {\_bgroup
2358 \_catcode`##=13 \_bgroup\_lccode`~=`## \_lowercase{\_egroup\_let~}=\_processline
2359 \_input{#1}%
2360 \_egroup
2361 }
2362 \_let\FormattedBook=\_ignoreit % for backward compatibility
2363 \_let\CommentedBook=\_ignoreit % for backward compatibility

```

23 Setting active character and \outer macros

Active character < used for references.

opbible.opm

```

2372 \_outer\_def\Note {\_Note}
2373 \_outer\_def\ww {\_ww}
2374 \_outer\_def\ChapterPre {\_ChapterPre}
2375 \_outer\_def\ChapterPost {\_ChapterPost}
2376 \_outer\_def\BookTilte {\_BookTitle}
2377
2378 \_def\_afterload{\_adef<{\_bref}}
2379 \_afterload
2380
2381 \_endnamespace

```

24 Index

<code>\.AddNote</code> 6	<code>\Cite</code> 22	<code>\fmtprepoet</code> 9
<code>\.addpre</code> 9	<code>\.Cnotetext</code> 8	<code>\fmtrepl</code> 10
<code>\alist!</code> 5, 10	<code>\cnvtext</code> 10	<code>\ftmadd</code> 9
<code>\amark</code> 3	<code>\.createlink</code> 14	<code>\.fullvref</code> 6
<code>\arrowtext</code> 27	<code>\.currbook</code> 3, 5, 10	<code>\.fullvrefm</code> 6
<code>\Article</code> 19	<code>\.currchapnum</code> 10	<code>\genbooks</code> 29
<code>\.begblock</code> 25	<code>\.currverse</code> 10	<code>\.gentovref</code> 5
<code>\begcenter</code> 9	<code>\.currversenum</code> 10–11	<code>\.hboxorllap</code> 9, 11
<code>\begChiasm</code> 26	<code>\.currversetext</code> 10	<code>\ind</code> 9, 11
<code>\bex!</code> 3–4	<code>\.doArticle</code> 19	<code>\insertBot</code> 24
<code>\.bibleinput</code> 3	<code>\.doCNote</code> 8	<code>\insertCite</code> 22
<code>\bibname</code> 4	<code>\.doImage</code> 18	<code>\insertSpanImage</code> 20–21
<code>\bmark</code> 3–4	<code>\.doNote</code> 6–7	<code>\insertSpanText</code> 21
<code>\BookException</code> 3–4	<code>\.dotopCite</code> 22	<code>\introfile</code> 3, 25
<code>\BookPost</code> 3–4	<code>\.endblock</code> 25	<code>\.iscolonin</code> 2, 7
<code>\BookPre</code> 3–4	<code>\.endbot</code> 18–19	<code>\.isdivisin</code> 2, 7
<code>\BookTitle</code> 4, 29	<code>\endcenter</code> 9	<code>\.isspacein</code> 2
<code>\.botins</code> 18	<code>\endChiasm</code> 26	<code>\l</code> 27
<code>\.botinsert</code> 18–19	<code>\.ensuredest</code> 14	<code>\.linkfspec</code> 12–13, 15
<code>\.botTitle</code> 18	<code>\f!</code> 4	<code>\.linkfspecone</code> 13
<code>\bpo!</code> 3–4	<code>\.filecontent</code> 29	<code>\.linklog</code> 13
<code>\bpr!</code> 3–4	<code>\filegen</code> 29	<code>\.linkpre</code> 12, 15
<code>\.bref</code> 11	<code>\.filename</code> 29	<code>\.linktext</code> 12–13
<code>\.brefBookChapter</code> 4	<code>\fmtfile</code> 3	<code>\.lqqbox</code> 22
<code>\.brefL</code> 13	<code>\fmtfont</code> 10	<code>\.ltextB</code> 12
<code>\.btit</code> 3	<code>\fmtins</code> 5, 9	<code>\.ltextC</code> 12
<code>\btit!</code> 4	<code>\fmtkeep</code> 5, 10	<code>\.ltextF</code> 12
<code>\.btitle</code> 29	<code>\.fmtpoetA</code> 9	<code>\.ltextN</code> 12
<code>\.buff</code> 5, 7, 9–11	<code>\.fmtpoetB</code> 9	<code>\.ltextP</code> 12
<code>\c</code> 26	<code>\.fmtpoetC</code> 9	<code>\.ltextS</code> 12
<code>\centeringmode</code> 9	<code>\fmtpoetry</code> 9, 11	<code>\.ltextV</code> 12
<code>\.chapafter</code> 11	<code>\fmtpre</code> 9, 11	<code>\megrednotes</code> 8
<code>\.chapbefore</code> 11	<code>\.fmtprebuf</code> 11	<code>\.myaddto</code> 2
<code>\.checknochapbooks</code> 4	<code>\.fmtprebuff</code> 9	<code>\.newaction</code> 5–6, 9, 18–19, 21

<code>\.newbook</code>	3-4	<code>\.printCnote</code>	8	<code>\shadowedtext</code>	25
<code>\.newlinkB</code>	13	<code>\printedbooks</code>	2	<code>\shadowparameter</code>	25
<code>\.nextww</code>	6, 16	<code>\.printintro</code>	25	<code>\swapCites</code>	22
<code>\.nextwwA</code>	6, 16	<code>\.printverse</code>	10-11	<code>\switch</code>	16
<code>\nochapbooks</code>	4	<code>\.printwarn</code>	2	<code>\.switchA</code>	16
<code>\noreduceref</code>	14	<code>\processbooks</code>	2, 4	<code>\.switchD</code>	16
<code>\normalchapnumbers</code>	28	<code>\.processline</code>	10	<code>\.switchN</code>	16
<code>\Note</code>	5-8, 16	<code>\.processverse</code>	10	<code>\timeline</code>	27
<code>\.NoteB</code>	6-7	<code>\.punctpword</code>	8	<code>\timelinewidth</code>	27
<code>\.notefail</code>	7	<code>\putArticle</code>	19	<code>\tline</code>	28
<code>\.noteins</code>	17-18	<code>\putBot</code>	24	<code>\tlines</code>	28
<code>\.noteinsert</code>	17	<code>\putImage</code>	18	<code>\tlput</code>	28
<code>\.notelog</code>	9	<code>\putSpanImage</code>	21	<code>\town</code>	26
<code>\.notenun</code>	6	<code>\putSpanText</code>	21	<code>\townparams</code>	26
<code>\notepre!</code>	6-7	<code>\putstext</code>	25	<code>\tracinglinks</code>	14
<code>\noteref!</code>	6	<code>\pword!</code>	6	<code>\tracingouterlinks</code>	14
<code>\noterule</code>	18	<code>\re</code>	14	<code>\.transformword</code>	6
<code>\.noteset</code>	17	<code>\.reducelinktext</code>	14	<code>\.trychapnote</code>	11
<code>\notesfile</code>	3	<code>\reduceref</code>	14	<code>\.upcasefirst</code>	7-8
<code>\notetext!</code>	6	<code>\renum</code>	6, 12-13, 17	<code>\v!</code>	16
<code>\notracinglinks</code>	13-14	<code>\renumlabel</code>	6-7	<code>\variants</code>	15, 17
<code>\.numvariants</code>	15	<code>\.renumlinktext</code>	13	<code>\.varnum</code>	16-17
<code>\Outline</code>	27	<code>\.renumvref</code>	6	<code>\vdef</code>	15
<code>\pbook!</code>	2	<code>\.replbuff</code>	5	<code>\.vdefB</code>	15
<code>\pg</code>	14-15	<code>\.replpre</code>	5	<code>\.versedef</code>	12
<code>\.prebuff</code>	8	<code>\.replprepost</code>	5	<code>\ww</code>	6, 16
<code>\.prelinkB</code>	13	<code>\rightnote</code>	27	<code>\x</code>	16
<code>\.prelinkC</code>	13	<code>\.rqqbox</code>	22	<code>\xA</code>	16
<code>\.prevnotepre</code>	7	<code>\.sedef</code>	2, 15	<code>\.Xdest</code>	14
<code>\.printbeforefirst</code>	11	<code>\.setheadline</code>	4		
<code>\.printchapnote</code>	11	<code>\.setvarnum</code>	17		