

OpBible – Technical Documentation

Version: 0.32, 2023-11-12

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	6
8	Inserting data from format files	9
9	Printing verses from <code>.txs</code> files	10
10	Bible references	12
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	30
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.

Finally, the `\processbooks` macro runs `\.finalwork`.

opbible.opm

```

132 \_def\.processbooks {\_par
133   \_ifx\tmark\undefined \_def\tmark{none}\_fi
134   \.checknochapbooks
135   \_useit{\_ea\.processbooksA \printedbooks} {}
136   \_useit{\_ea\.processbooksB \printedbooks} {}
137   \.finalwork
138 }
139 \_def\.processbooksA #1 {%
140   \_if\_relax#1\_relax \_else \_sxddef{pbook!#1}{\_ea\.processbooksA \_fi
141 }
142 \_def\.processbooksB #1 {%
143   \_if\_relax#1\_relax \_else
144     \_edef\amark{#1}
145     \_edef\bmark{\_cs{f!#1}}
146     \_edef\.btit{\_cs{btit!#1}}
147     \_begingroup
148       \_edef\.currbook{#1}
149       \.newbook{#1}
150       \_wterm{^^J** \_cs{btit!#1} {#1} (\string\tmark: \tmark) **^^J}
151       \_cs{bex!#1}
152       \_isfile{\introfile}\_iftrue \.printintro
153       \_else \.printwarn{File with introduction text \introfile\_space not found}\_fi
154 %
155       \_isfile{\fmtfile}\_iftrue \_input{\fmtfile}
156       \_else \.printwarn{File with format info \fmtfile\_space not found}\_fi
157       \_isfile{\notesfile}\_iftrue \_input{\notesfile}
158       \_else \.printwarn{File with notes \notesfile\_space not found}\_fi
159       \_cs{bpr!#1}
160       \.bibleinput{\txsfile}
161       \.chapafter % material after the last chapter
162       \_cs{bpo!#1}
163     \_endgroup
164     \_ea \.processbooksB
165   \_fi
166 }
167 \_nspublic \processbooks ;

```

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

opbible.opm

```

173 \_def\.newbook#1{\_vfil\_supereject
174   \_let\prelinkB=\.currbook \.chapnum=0
175   \_def\prelinkC{0}\_def\prelinkV{0}
176   \_ea\iniheadline\_ea{\.btit}

```

```

177 \_line{\_hss\bookfont\btit\_hss}
178 \_label[cref!#1]\_wlabel{#1}
179 \_par\_nobreak\_medskip
180 }

```

`\.iniheadline{<book-title>}` sets `_headline` with delay (current page is without head line, next pages include headlines). It uses `\.setheadline{<book-title>}`. 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

```

191 \_def\.iniheadline#1{\_global\_headline={\_hfil \.setheadline{#1}}}
192 \_def\.setheadline#1{\_global\_headline={\_headfont
193 \_ifodd\_pageno
194 \_rlap{\_it\bibname\_hss}%
195 \_hfil \_the\_pageno\_hfil
196 \_hbox to\lrmargin{\_hss\_bf#1\_ifx^\_botmark^\_else\_space \_botmark\_fi}%
197 \_kern-\lrmargin
198 \_else
199 \_kern-\lrmargin
200 \_hbox to\lrmargin{\_bf#1 \_firstmark\_hss}%
201 \_hfil \_the\_pageno\_hfil
202 \_llap{\_hss\_it\bibname}%
203 \_fi
204 }
205 }
206 \_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

```

219 \_def\.checknochapbooks {%
220 \_ifx\nochapbooks\_undefined
221 \_printwarn{\_noexpand\nochapbooks (boks without chapters) undefined.}%
222 \_def\nochapbooks{}%
223 \_else \_edef\nochapbooks{\_space\nochapbooks\_space}\_fi
224 }

```

`\.finalwork` runs end game when all books are printed.

opbible.opm

```

230 \_def\.finalwork{
231 \_wterm{^^J=== Total \_csstring\Note's number = \_the\notenum.^^J}
232 }

```

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\btit!<a-mark>{<title>}`, `\def\fi!<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

```

249 \_def\genbooks{}
250 \_def\BookTitle #1 #2 #3{%
251 \_sxddef{btit!#1}{#3}\_sxddef{f!#1}{#2}\_sxddef{fb!#2}{#1}%
252 \_addto\genbooks{#2 }%
253 }

```

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.

```

265 \_outer\_long\_def\BookException #1 #2{\myaddto{bex!#1}{#2}}
266 \_outer\_long\_def\BookPre      #1 #2{\myaddto{bpr!#1}{#2}}
267 \_outer\_long\_def\BookPost    #1 #2{\myaddto{bpo!#1}{#2}}
268
269 \_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.

```

277 \_long\_def\ChapterPre #1{\_def\chapbefore{#1}}
278 \_long\_def\ChapterPost #1{\_def\chapafter{#1}}
279
280 %\_outer\_def\ChapterPre {\_ChapterPre}
281 %\_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.

```

301 \_def\.newaction#1#2{%
302   \_unless\_ifcname alist!#1\_endcname \_sdef{alist!#1}{\_fi
303   \_ea\_addto\_csname alist!#1\_endcname{#2}%
304 }

```

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`.

```

315 \_def\.replpre#1#2#3{%
316   \_ifx^#2^\_def\.tmp{#1}{\_ea\_ea\_ea\_def\_ea\_ea\_ea\.buff\_ea\_ea\_ea{\_ea\.tmp\.buff}}%
317   \_else
318     \.replbuff{#2}{#1{#2}}{#3}%
319   \_fi
320 }

```

`\.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.

```

331 \_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.

```

341 \_def\.replbuff #1#2#3{%
342   \_def\.replpredo##1#1##2\_end{%
343     \_ifx\_end##2\_end \_def\.text{#1}#3% <fail>
344     \_else \.replsave ##1#2##2\_end \_fi
345   }%
346   \_def\.replsave##1#1\_end{\_def\.buff{##1}}%
347   \_ea\.replpredo\.buff#1\_end
348 }

```

7 The \Note macro

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

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

opbible.opm

```
362 \_def\gentovref#1{\currbook/\gentovrefA#1-\end}
363 \_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

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

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

opbible.opm

```
381 \_def\transformword#1{%
382   \_ifcsname v!\tmark!#1\_endcsname \_lastnamedcs
383   \_else #1\_fi
384 }
```

`\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<gen-vref> $\langle space \rangle$ { $\langle word \rangle$ }= { $\langle pword \rangle$ } $\langle text \rangle$ \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 svese 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. `\notenum` 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

```
430 \_newcount\notenum
431 \_def\Note #1 #2{%
432   \_edef\fullvref{\gentovref{#1}}%
433   \_ea\isversezero\fullvref\_iftrue
434     \_ea\NoteB
435   \_else
436     \_incr\notenum
437     \_edef\fullvrefm{\_ea\renumvref\fullvref\_relax}%
438     \_def\tmp{#1}\sedef{notepre!\_the\notenum}{\_ea\renumlabel\fullvrefm\_relax}%
```

```

439     \_ifx\.\nextww\_undefined
440         {\_def\.\printwarn##1{\_xdef\.\tword{\.\transformword{#2}}}%
441         \_else \_xdef\.\tword{\.\nextww}\_fi
442         \_afterfi{\_isnextchar={\.\NoteA}{\.\NoteA={}}}%
443     \_fi
444 }
445 \_def\.\NoteA=#1#2% #2 separated by \par or \_par:
446
447 {%
448     \_sdef{notetext!\_the\.\notenum}{\_ignorespaces#2}%
449     \_sedef{noteref!\_the\.\notenum}{\.\fullvrefm}%
450     \_ifx\.\nextww\_undefined
451         \_ifx^#1^\_sdef{pword!\_the\.\notenum\_ea}\_ea{\.\tword}\_else \_sdef{pword!\_the\.\notenum}{#1}\_fi
452     \_else
453         \_sdef{pword!\_the\.\notenum\_ea}\_ea{\.\nextwwA}%
454         \_let\.\nextww=\_undefined \_let\.\nextwwA=\_undefined
455     \_fi
456     \.reducetword
457     \_ea\.\addNote\_\expanded{{\.\fullvrefm}{\_the\.\notenum}{\.\tword}}%
458 }
459 \_def\.\addNote#1#2#3{%
460     \_ifx^#3^\_tword is empty
461         \_edef\.\tmp{\_cs{notepre!#2}}%
462         \_ea \.\isdivisin\.\tmp-\_iftrue
463             \.\newaction{#1}{\.\replpre{\.\doNote{#2}}{}}}%
464         \_else
465             \.\newaction{#1}{\_\addto\.\prebuff{\.\doCNote{#2}}{}}}%
466         \_fi
467     \_else
468         \.\newaction{#1}{\.\replpre{\.\doNote{#2}}{#3}{\.\notefail{#2}}}%
469     \_fi
470 }
471 %\_\outer\_def\Note{\.\Note} % will be done at the end of this macro file

```

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

opibble.opm

```

480 \_def\.\NoteB #1% #1 separated by \par or \_par
481
482 {%
483     \_sdef{chapnote!\.\fullvref}{\_ignorespaces#1}%
484 }
485 \_def\.\isversezero#1/#2:#3\_iftrue{\_ifnum #3=0 }

```

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

opibble.opm

```

498 \_def\.\renumlabel#1/#2\_relax#2%
499     \_ea\.\isdivisin\.\tmp-\_iftrue --\_ea\.\renumlabelA\.\tmp\_relax#2\_relax \_fi
500 }
501 \_def\.\renumlabelA#1:#2-#3\_relax#4:#5\_relax{%
502     \.\iscolonin#3:\_iftrue #3\_else \_the\_\numexpr#5+#3-#2\_relax \_fi
503 }

```

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

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

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

opibble.opm

```

518 \_def\.\notefail#1{%
519     \.\printwarn{\_csstring\Note: \.\currverse: The text "\_unexpanded\_ea{\.\text}" not found}%
520     \.\replpre{\.\doNote{#1}}{}}% \Note is registered with the beginning of the verse
521 }

```


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

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

opbible.opm

```
533 \_def\.prevnotepre{}
534 \_def\.doNote#1#2{%
535   \_edef\.tmpb{\_cs{notepre!#1}}%
536   \.notelog{\_space\_space \_csstring\\Note \.tmpb\_space {#2}={\_cs{pwd!#1}} (#1)}%
537   \.noteinsert{%
538     {\_bf \_ifx\.prevnotepre\.tmpb \_else \.tmpb \_enskip \_glet\.prevnotepre=\.tmpb \_fi
539     \.trymakedest{n:\_cs{noteref!#1}}%
540     \_edef\.tmpb{\_csname pwd!#1\_endcsname}%
541     \_ifx\.tmpb\_empty \_else
542       \_addto\.tmpb{.\_relax}\.punctpwd
543       \_ea\.upcasefirst \.tmpb\_space
544       \_fi
545     }% end of \bf
546     \_cs{notetext!#1}}%
547     {\notecolor#2}}%
548   }
549 \_def\_printfnotemark{}
550 \_def\_textindent#1{\_noindent}
```

The `<pwd>` is typically all lowercase. But we want to capitalize the first letter of the `<pwd>` when printing by `\.upcasefirst`. You can say `\let\.upcasefirst=\relax` if you don't want this feature.

opbible.opm

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

The dot is added to `<pwd>` when it is printed. But if `<pwd>` ends by ! or ? or . then the added dot is ugly. We have to correct it in the `\.punctpwd` macro. Note that `<pwd>` is saved to `\.tmpb`.

opbible.opm

```
568 \_def\.punctpwd{\_replstring\.tmpb{!.\_relax}{!}\_replstring\.tmpb{?.\_relax}{?}}%
569 \_replstring\.tmpb{.\_relax}{.}}
```

When `\Note` has empty parameter `<word>` (i.e. `<tword>`) 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 `\.doCNote{<notenum>}` is run from `\.prebuff` and it only adds the text of the note to the `\.Cnotetext` buffer. When `\.prebuff` is completed then `\.printCnote` prints the merged note.

opbible.opm

```
580 \_def\.doCNote #1{%
581   \_edef\.tmpb{\_csname pwd!#1\_endcsname}%
582   \.notelog{\_space\_space \_csstring\\Note \.tmpb\_space }={\_cs{pwd!#1}} (#1)}%
583   \_edef\.prevnotepre{\_cs{notepre!#1}}%
584   \_ifx\.tmpb\_empty \_else
585     \_addto\.tmpb{.\_punctpwd
586     \_edef\.tmpb{{\_noexpand\_bf \_ea\.upcasefirst\.tmpb\_noexpand~}}%
587     \_ea\_addto \_ea\.Cnotetext \_ea{\.tmpb}%
588     \_fi
589     \_ea\_ea\_ea\_addto\_ea\_ea\_ea\.Cnotetext\_ea\_ea\_ea{\_csname notetext!#1\_endcsname}%
590   }
591 \_def\.printCnote{%
592   \_ifx\.Cnotetext\_empty \_else
593     \.noteinsert{%
594       {\_bf \_ea\.nobook\.currverse\_relax \.trymakedest{n:\.currverse}} \.Cnotetext
595     }%
596     \_fi
597   }
598 \_def\.nobook #1/#2\_relax {#2} % only chapter:verse is printed
```

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

opbible.opm

```
607 \_def\.reducetword{}
608 \_def\.mergednotes{\_def\.reducetword{\_def\.tword{}}}
609 \_nspublic \mergednotes ;
```


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

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

opbible.opm

```
622 \_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

```
639 \_def\.fmtpre#1#2{\.newaction{\.gentovref{#1}}{\_addto\.fmtprebuff{#2}}}
640 \_def\.fmtpreind#1#2{\.newaction{\.gentovref{#1}}{\.addpre\.preindbuff{#2}}}
641 \_def\.fmtadd#1#2{\.newaction{\.gentovref{#1}}{\_addto\.buff{\_empty#2}}}
642 \_def\.fmtins#1#2#3{\.newaction{\.gentovref{#1}}{\.replprepost{#2}}{\_empty#3}{\.fmtfail{#3}}}
643 \_def\.fmtfail#1{\.fmtwarn\_addto\.fmtprebuff{#1}}
644 \_def\.fmtwarn{\.printwarn{\_string\fmtins: \.currverse: The text "\.text" not found}}
645 \_def\.addpre#1#2{\_ea\.addpreA \_ea{#1}{#2}#1}
646 \_def\.addpreA #1#2#3{\_def#3{#2}#1}
647
648 \_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

```
657 \_newdimen\centermargin \centermargin=4em
658 \_def\.begcenter{\_par \_ifnum\_lastpenalty<10000 \_medskip \_fi
659 \_bgroup
660 \_def\.centeringmode{y}
661 \_parindent=0pt
662 \_leftskip=\centermargin plusifill
663 \_rightskip=\_leftskip
664 }
665 \_def\.endcenter{\_par
666 \_ifx\.centeringmode\_undefined
667 \.printwarn{\_noexpand\endcenter ignored: no \_noexpand\begcenter precedes}
668 \_else \_egroup \_medskip \_fi
669 }
670 \_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 lapped after `\ind`.

opbible.opm

```
681 \_newifi\_ifopb_firstverse
682
683 \_def\.ind#1{\_unless \_ifopb_firstverse \_par \_else \_hskip-\_parindent \_fi
684 \_noindent
685 \_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}`.

```

695 \_def\fmtpoetry#1#2{\_def\.tmpa{#1}\fmtpoetA #2\_end}
696 \_def\fmtpoetA #1/{\_def\.tmpb{#1}\_tmpnum=1 \fmtpoetB}
697 \_def\fmtpoetB #1{\_ifx/#1 \_incr\_tmpnum \_ea\fmtpoetB \_else \_afterfi{\_fmtpoetC#1}\_fi}
698 \_def\fmtpoetC #1{%
699 \_expanded{\_ifx\.tmpb\_empty \_noexpand\fmtpreind{\_tmpa}\_else
700 \_noexpand\fmtins{\_tmpa}{\_tmpb}\_fi{ \_noexpand\_ind{\_the\_tmpnum}}}%
701 \_ifx\_end#1 \_else \_afterfi{\_fmtpoetA#1}\_fi
702 }
703 \_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 $\langle whom \rangle$.

```

714 \_def\fmtfont#1#2#3{%
715 \_newaction{\_gentovref{#1}}{\_replprepost{#2}{\bgroup#3}{\egroup}{\_fmtwarnf\fmtfont}}
716 \_def\fmtkeep#1#2{%
717 \_newaction{\_gentovref{#1}}{\_replpre{#2}{\_fmtwarnf\fmtkeep}}
718 \_def\fmtrepl#1#2#3{\_newaction{\_gentovref{#1}}{\_replbuff{#2}{#3}{\_fmtwarnf\fmtkeep}}
719
720 \_def\fmtwarnf#1{\_printwarn{\_string#1: \_currverse: The text "\_text" not found}}
721 \_nspublic \fmtfont \fmtkeep \fmtrepl ;

```

9 Printing verses from .txs files

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

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

```

734 \_eoldef\processline#1{\_processverse \currbook/#1\_end}

```

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

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

```

749 \_newcount\chapnum
750 \_def\processverse #1 #2\_end{%
751 \_xdef\currverse{#1}%
752 \_preparechapverse #1
753 \_let\prelinkV=\currversenum
754 \_gdef\buff{#2}\_gdef\fmtprebuff{\_gdef\preindbuff{\_gdef\prebuff{\_gdef\Cnotetext{}}%
755 \_ifx\verseto\_empty \_csname alist!#1\_endcsname \_else
756 \_fornum \_versefrom..\verseto \_do{\_csname alist!\currbook/\currchapnum:#1\_endcsname}%
757 \_fi
758 \_ifnum\currchapnum=\chapnum \_else
759 \_ifnum\chapnum>1 \chapafter \_fi
760 \_let\prelinkC=\currchapnum \chapnum=\currchapnum\_relax
761 \chapbefore
762 \_label[cref!\currbook\_space\_the\chapnum]\_wlabel{\currbook~\_the\chapnum}%
763 \_fi
764 \printverse
765 }
766 \_def\preparechapverse #1/#2:#3 {\_def\currchapnum{#2}%
767 \_def\verseto{}}%
768 \_isdivisin #3-\_iftrue \_defversefromto #3\_end
769 \_else \_def\currversenum{#3}\_glet\currversetext=\currversenum
770 \_fi
771 }
772 \_def\defversefromto #1-#2\_end{%
773 \_def\versefrom{#1}\_def\verseto{#2}%
774 \_def\currversenum{#1}\_gdef\currversetext{#1--#2}}

```

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

opbible.opm

```
782 \_def\prepareversetext{}
783 \_def\cnvtext#1#2{\_addto\prepareversetext{\_replstring\buff{#1}{#2}}
784 \_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 immediatelly after `\ind` is used. The `\.hboxorllap` macro does this game.

opbible.opm

```
800 \_def\.printverse{%
801 \.fmtprebuff % material accumulated by \fmtpre
802 \_ifnum\.currversenum=1 \.firstversetrue \.printbeforefirst \_fi
803 \_quitmode \_mark{\.currchapnum:\.currversetext}%
804 \_ifx\verseto\_empty \.trymakedest{v:\.currverse}%
805 \_else \_fornum \.versefrom..\verseto \_do{%
806 \.trymakedest{v:\.currbook/\.currchapnum:##1}}%
807 \_fi
808 \.preindbuff
809 \_raise5pt\.hboxorllap{\_unless\_ifnum\.currversenum=1 \.markfont\.currversetext\,\_fi}%
810 \.firstversefalse
811 \.prepareversetext
812 \.prebuff\.printCnote\buff \_space
813 }
814 \_def\.hboxorllap{\_ifnum\_spacefactor=1001 \_ea\_llap \_else \_ea\_hbox \_fi}
815
816 \_def\.printbeforefirst{%
817 \_par\_nobreak \_medskip
818 \.trychapnote
819 \_setbox0=\_vtop{\_kern-1.5ex \_ewref\_sxdef{{ch!\.currbook/\_the\chapnum}{\_string\mypage}}
820 \_hbox{\_setfontsize{at50pt}\_bf\LiRed\_the\chapnum}}
821 \_dp0=0pt
822 \_tmpdim=\_lrmargin
823 \_advance\_tmpdim by4pt
824 \_ifnum\_the\chapnum>9 \_advance\_tmpdim by19pt \_fi
825 \_ifodd\_trys{ch!\.currbook/\_the\chapnum}{0}
826 \_moveright\_tmpdim \_line{\_hss\_box0}
827 \_else \_moveleft\_tmpdim \_box0 \_fi
828 \_nobreak \_vskip-\_medskipamount
829 \_nobreak \_nointerlineskip \_noindent
830 }
```

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

opbible.opm

```
837 \_def\.trychapnote{%
838 \_ifcsname chapnote!\.currbook/\_the\chapnum:0\_endcsname
839 \.printchapnote{\_cs{chapnote!\.currbook/\_the\chapnum:0}}\_fi
840 }
841 \_def\.printchapnote #1{\_par
842 {\_leftskip=\_parindent plus1fill \_rightskip=\_leftskip \_noindent\_it #1\_par}
843 \_medskip
844 }
845 \_nspublic \printchapnote ;
```

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

opbible.opm

```
853 \_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 `<from>-<to>` format is given
- `\.ltextN` ... the `<to>` part from the `<from>-<to>` format.

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

opbible.opm

```

881 \_def\.linktext{\.ltextP\.ltextB\.ltextC\.ltextV\.ltextS\.ltextF\.ltextN}
882 \_def\.bref #1>{\_let\.brefH=\_relax \_def\.linkspec{#1}\_isnextchar"{\.brefA}{\.brefA"}#1>}
883 \_def\.brefA"#1"{\_def\.ltextP{#1}%
884 \_isnextchar{ }{\_addto\.ltextP{~}\_afterassignment\.brefB\_let\.next= }%
885 {\_isnextchar_{}\_def\.brefH{ }\_afterassignment\.brefB\_let\.next= }{\.brefB}}%
886 }
887 \_def\.brefB #1>{% #1 is link-spec
888 \_def\.ltextB{ }\_def\.ltextC{ }\_def\.ltextF{ }\_def\.ltextN{ }%
889 \_isspacein #1 \_iftrue
890 \_iscolonin #1:\_iftrue \.brefBookChapterVerse #1>%
891 \_else \.brefBookChapter #1>\_fi
892 \_else \_iscolonin #1:\_iftrue \.brefChapterVerse #1>%
893 \_else \.brefVerse #1>%
894 \_fi\_fi
895 \_def\.linkpre{v}%
896 \_isnextchar n{\_def\.linkpre{n}\.brefC}%
897 {\_isnextchar g{\_def\.linkpre{g}\.brefC}%
898 {\_isnextchar a{\_def\.linkpre{a}\.brefC}%
899 {\_isnextchar i{\_def\.linkpre{i}\.brefC}{\.brefD}}}}%
900 }
901 \_def\.brefC{\_afterassignment\.brefD \_let\.next= }
902
903 \_def\.brefBookChapterVerse #1 #2:#3>{\_def\.ltextB{#1-}\.brefChapterVerse #2:#3>}
904 \_def\.brefBookChapter #1 #2>{\_def\.ltextB{#1-}%
905 \_isinlist\nochapbooks{ #1 } \_iftrue
906 \_def\.ltextC{ }\_let\.ltextCin=\.ltextnCin \_afterfi{\.brefVerse #2>}%
907 \_else \_afterfi{\.brefChapter #2>}\_fi}
908 \_def\.brefChapterVerse #1:#2>{\_def\.ltextC{#1-}\.brefVerse #2>}
909 \_def\.brefVerse #1>{%
910 \_isdivisin #1-\_iftrue \.brefFromTo #1>%
911 \_else \.versedef#1\_relax\_fi
912 }
913 \_def\.brefChapter #1>{%
914 \_isdivisin #1-\_iftrue \.brefFromTo #1>\_let\.ltextC=\.ltextV
915 \_else \_def\.ltextC{#1}\_fi
916 \_def\.ltextV{ }\_def\.ltextS{ }%
917 }
918 \_def\.brefFromTo #1-#2>{\.versedef#1\_relax\_def\.ltextF{--}\_def\.ltextN{#2}}

```

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

opbible.opm

```

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

```

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

```

934 \_def\.\brefD{%
935   \_ifnum 0\.\ltextV=0 \_def\.\ltextV{\}_\fi
936   \_if a\.\linkpre \_ifx\.\ltextV\_empty \_else \_edef\.\ltextC{\.\ltextV:}\_def\.\ltextV{\}_\fi\_fi
937   \_edef\.\linkfspec{\_ea\.\ltextBin\.\ltextB-/\_ea\.\ltextCin\.\ltextC:/\_ea\.\ltextVin\.\ltextV:/}%
938   \.\brefL
939 }
940 \_def\.\ltextBin #1:#2/{\_ifx^#1^\.\prelinkB \_else #1\_immediateassignment\_def\.\prelinkB{#1}\_fi}/
941 \_def\.\ltextCin #1:#2/{\_ifx^#1^\.\prelinkC \_else #1\_immediateassignment\_def\.\prelinkC{#1}\_fi:}
942 \_def\.\ltextVin #1:#2/{\_ifx^#1^\.\prelinkV \_else #1\_immediateassignment\_def\.\prelinkV{#1}\_fi}
943 \_def\.\ltextnCin #1:#2/{1:\_immediateassignment\_let\.\ltextCin=\.\ltextsCin}
944 \_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.

```

954 \_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

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

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

```

978 \_def\.\brefL{%
979   \_edef\.\linkspecm{\_ea\.\renumvref\.\linkfspec\_relax}%
980   \_ifx\.\linkfspec\.\linkspecm \_else
981     \_ea\_ea\_ea\.\renumlinktext \_ea\.\linkfspec \_ea\_relax \.\linkspecm \_relax
982     \_let\.\linkfspec=\.\linkspecm
983   \_fi
984   \_ifx\.\ltextV\_empty \_ifx\.\ltextC\_empty \_else \_ea\.\linkfspecone \.\linkfspec\_end \_fi\_fi
985   \_if a\.\linkpre\_relax \_ea\.\linkspecarticle \.\linkfspec\_end \_fi
986   \_if i\.\linkpre\_relax \_ea\.\linkspecintro \.\linkfspec\_end \_fi
987   \_ifx \.\ltextB\_empty \_else \_ea \.\newltextB \.\ltextB \_fi
988   \.\reducelinktext
989   \.\linklog{\.sspace <\_unexpanded\_ea{\.\linkspec}>\.\linkpost = [\.\linkpre:\.\linkfspec]%
990     {\_ifx\.\brefH\_empty \.\ltextP \_else \.\linktext\_fi}}%
991   \.\ensuredest \.\createlink
992 }
993 \_def\.\linkfspecone #1:#2\_end {\_def\.\linkfspec{#1:1}\_def\.\prelinkV{1}}
994 \_def\.\linkspecarticle #1/#2:#3\_end {\_def\.\linkfspec{#1/#2}}
995 \_def\.\linkspecintro #1/#2\_end {\_def\.\linkfspec{#1/}}
996
997 \_def\.\renumlinktext #1/#2:#3\_relax #4/#5:#6\_relax{%
998   \_ifx\.\ltextC\_empty \_else \_def\.\ltextC{#5:}\_fi
999   \_def\.\ltextV{#6}%
1000   \_ifx\.\ltextN\_empty \_else
1001     \_ifx\.\ltextF\.\ltextDD
1002       \_isinlist\.\ltextN{:}\_iftrue
1003       \_ifcsname rn!\tmark!#1/\.\ltextN\_endcsname \_edef\.\ltextN{\_cs{rn!\tmark!#1/\.\ltextN}}%
1004       \_fi
1005       \_else \_edef\.\ltextN{\_the\_numexpr#6+\.\ltextN-#3\_relax}\_fi
1006     \_else \_let\.\tmp=\_ignoreit % \.\ltextN is a list of verses, for example 7,9,13
1007     \_ea\_foreach\.\ltextN,\_do #1,{\_edef\.\tmp{\.\tmp,\_the\_numexpr#6+##1-#3}}%
1008     \_let\.\ltextN=\.\tmp
1009   \_fi
1010   \_fi

```

```

1011 }
1012 \_def\ltextDD{--}
1013
1014 \_def\newltextB #1-{\_edef\ltextB{\_trycs{v!\_tmark!#1}{#1}~}}
1015
1016 \_def\sspace{\_space\_space\_space\_space}
1017 \_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

```

1026 \_def\.reducelinktextA{%
1027   \_edef\ltextB{\_currbook~}%
1028   \_ifx\ltextB\ltextB\_def\ltextB{}%
1029   \_edef\ltextC{\_trycs{\_opb\_currchapnum}{?}:}%
1030   \_ifx\ltextC\ltextC\_def\ltextC{}%
1031   \_fi\_fi
1032   \_ifcsname \_opb\_reA\_endcsname \_let\.reducelinktext=\_reA \_fi % after \re
1033 }
1034 \_def\.reduceref{\_let\.reducelinktext=\_reducelinktextA}
1035 \_def\.noreduceref{\_let\.reducelinktext=\_relax}
1036 \_noreduceref % default
1037
1038 \_def\.re{\_let\.reA=\_reducelinktext \_reduceref}
1039
1040 \_nspublic \reduceref \noreduceref \re ;

```

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

opbible.opm

```

1046 \_def\tracinglinks{\_let\linklog=\_wlog}
1047 \_def\notracinglinks{\_let\linklog=\_ignoreit}
1048 \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-existed books. By default, these links are not logged because we assume that no whole Bible is processed but only selected books.

opbible.opm

```

1062 \_def\.createlink{%
1063   \_ifx\ltextB\ltextB\_empty \_let\linktext=\_ltextP\_fi
1064   \_ea.isprintedbook\linkfspec \_iftrue
1065   \_link[\linkpre:\linkfspec]{\_ilinkcolor}{\linktext}%
1066   \_else {\_ilinkcolor\linktext}\_fi}%
1067 }
1068 \_def.isprintedbook #1/#2\_iftrue{\_ifcsname pbook!#1\_endcsname}
1069 \_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

```

1089 \_newwrite\ltextB
1090 \_immediate\_openout\ltextB=\_jobname.xrf
1091 \_openref
1092
1093 \_def\.ensuredest{\_immediate\_write\ltextB{\_string\_sdef{\linkpre:\linkfspec}{}}}
1094 \_refdecl{
1095   \_isfile{\_jobname.xrf}\_iftrue \_input{\_jobname.xrf}\_fi^^J

```



```

1096 \_def\Xdest#1{\_ifcsname pg:#1\_endcsname \_sxddef{pg:#1}{\_ea\_usesecond\_currrpage}\_fi}^^J
1097 \_def\mypage{\_ea\_usesecond\_currrpage}
1098 }
1099 \_def\trymakedest#1{%
1100 \_ifcsname #1\_endcsname \_dest[#1]\_ea\_glet\_csname #1\_endcsname \_undefined \_fi
1101 \_ewref\Xdest{#1}}%
1102 }

```

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

```

1110 \_def\pg{%
1111 \_ifcsname pg:\linkpre:\linkspec\_endcsname
1112 {\_edef\linktext{\_cs{pg:\linkpre:\linkspec}}\_let\ brefH=\_relax \_createlink}%
1113 \_else {\Red ??}\_fi
1114 \_immediate\_write\_xrf{\_string\_sdef{pg:\linkpre:\linkspec}{??}}%
1115 }
1116 \_nspublic \pg ;

```

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

```

1122 \_def\cref[#1]{\_ref[cref!#1]}
1123
1124 \_nspublic \cref ;

```

11 Language variants

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

```

1136 \_newcount\ numvariants
1137 \_def\variants{\_tmpnum=0 \_afterassignment\ variantsA \ numvariants}
1138 \_def\ variantsA{%
1139 \_ifnum\_tmpnum<\ numvariants
1140 \_advance\_tmpnum by1
1141 \_afterfi{\ variantsB{\_the\_tmpnum}}%
1142 \_fi
1143 }
1144 \_def\ variantsB#1#2{%
1145 \_ifnum#1=1 \_gdef\tmarkA{#2}\_sxddef{var!1}{#2}%
1146 \_else \_sxddef{var!#1}{#2}%
1147 \_fi
1148 \ variantsA
1149 }
1150 \_nspublic \variants ;

```

`\vdef` $\{\langle phrase-A \rangle\}$ $\{\langle phrase-B \rangle\}$ $\{\langle phrase-C \rangle\}$... does `\def\v!<tmark-B>!\langle phrase-A \rangle{\langle phrase-B \rangle}` `\def\v!<tmark-C>!\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{\langle number \rangle}{\langle param \rangle}` does real work and it defines (roughly sepaking):

```

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

```

```

1167 \_def\vdef#1{\_def\ tmp{#1}%
1168 \_ifcsname v!\_trycs{var!2}{!}\_tmp\_endcsname
1169 \_printwarn{\_noexpand\vdef used secondly for phrase {\_tmp}, ignored}\_fi
1170 \_tmpnum=1 \_ea\vdefA
1171 }
1172 \_def\vdefA{%
1173 \_ifnum\_tmpnum<\ numvariants
1174 \_advance\_tmpnum by1
1175 \_afterfi{\ vdefB{\_the\_tmpnum}}%
1176 \_fi
1177 }

```



```

1178 \_def\.\vdefB#1#2{\_def\.\tmpa{}}%
1179 \_ifx\.\vdef#2\_def\.\tmpa{#2}\_fi
1180 \_ifx\.\tmpa\_empty
1181 \_ifx^#2^\_else
1182 \_unless \_ifcstype v!\_cs{var!#1}!\_tmp\_endcstype
1183 \_sedef{v!\_cs{var!#1}!\_tmp}{\_ifx"#2\.\prevcs{#1}\_tmp \_else#2\_fi}%
1184 \_fi\_fi
1185 \_ea\.\vdefA
1186 \_else \_errmessage{\_string\vdef: too few parameters. To be read again: \_string#2}%
1187 \_ea\.\tmpa
1188 \_fi
1189 }
1190 \_def\.\prevcs #1#2{\_ifnum#1=2 #2\_else \_cs{v!\_cs{var!\_the\_numexpr#1-1\_relax}!#2}\_fi}
1191
1192 \_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*.

opbible.opm

```

1205 \_def\.\x/#1/{\_trycs{v!\tmark!#1}{\xA#1/}}
1206 \_def\.\xA#1/{#1\_ifx\tmarkA\_undefined \_else \_ifx\tmark\tmarkA \_else
1207 \_printwarn{\_string\x/#1/ -- this phrase is undefined by \_csstring\vdef}%
1208 \_fi\_fi
1209 }
1210 \_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.

opbible.opm

```

1223 \_def\.\ww{%
1224 \_ifx\.\varnum\_undefined \_setvarnum \_fi
1225 \_tmpnum=0
1226 \_ifx\.\nextww\_undefined \_ea\.\wwA
1227 \_else \_printwarn{Only single \_csstring\ww must be before \_csstring\Note}%
1228 \_ea\.\wwB \_fi
1229 }
1230 \_def\.\wwA#1#2 {\_advance\_tmpnum by1
1231 \_isequal{"}{#1}\_iffalse
1232 \_def\.\nextww{#1}\_def\.\nextwwA{#2}%
1233 \_ifx\.\nextwwA\_empty \_let\.\nextwwA=\.nextww \_else \_ea \_redefwwA #2\_end \_fi
1234 \_fi
1235 \_ifnum\.\varnum=\_tmpnum \_ifnum\_tmpnum<\.numvariants \_ea\_ea\_ea \_wwB \_fi
1236 \_else \_ea \_wwA \_fi
1237 }
1238 \_def\.\wwB#1 {\_advance\_tmpnum by1
1239 \_ifnum\_tmpnum<\.numvariants \_ea\.\wwB \_fi
1240 }
1241 \_def\.\redefwwA =#1\_end{\_def\.\nextwwA{#1}}
1242
1243 % \_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.

opbible.opm

```

1257 \_newtoks\.\switchD
1258 \_def\.\switch {\_let\.\switchN=\.switchA \_suppressoutererror=1 \.switchN}

```

```

1259 \_long\_def\switchA #1#2{\switchD={#2\_let\switchN=\switchI}%
1260 \_ifx\_relax#1\_relax \_the\switchD
1261 \_else \_foreach #1,\_do ##1,{\_def\tmp{##1}\switchC}%
1262 \_fi
1263 \_futurelet\_next\switchB
1264 }
1265 \_def\switchB{\_ifx\_next\_bgroup \_ea\switchN \_else \_suppressoutererror=0 \_fi}
1266 \_long\_def\switchI #1#2{\_futurelet\_next\switchB}
1267 \_def\switchC{\_ifx\tmp\tmark \_the\switchD \_fi}
1268
1269 \_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

```

1277 \_def\setvarnum{\_gdef\varnum{0}%
1278 \_ifnum\numvariants=0 \_gdef\varnum{1}\_wlog{There is only single language variant (1)}%
1279 \_else
1280 \_tmpnum=0
1281 \_loop
1282 \_advance\_tmpnum by1
1283 \_ea\_ifx \_csname var!\_the\_tmpnum\_endcsname \tmark \_xdef\varnum{\_the\_tmpnum}\_fi
1284 \_ifnum\_tmpnum<\numvariants \_repeat
1285 \_ifnum \varnum=0 \_errmessage{\_noexpand\tmark isn't set, \_noexpand\setvarnum failed}%
1286 \_else \_wlog{Language variant set by \_string\tmark{\tmark} (\varnum)}\_fi
1287 \_fi
1288 }

```

`\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

```

1302 \_def\renum #1 #2:#3 = #4 #5:#6-#7 {%
1303 \_tmpnum=#3\_relax
1304 \_fornum #6..#7 \_do {\_sxddef\rn!#4!#1/#2:\_the\_tmpnum}{#5:##1}\_incr\_tmpnum}%
1305 }
1306 \_nspublic \renum ;

```

12 Inserting notes to the page

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

opbible.opm

```

1315 \_newinsert \noteins
1316 \_skip\noteins=\bigskipamount % noterule height
1317 \_count\noteins=500 % two columns
1318 \_dimen\noteins=\maxdimen % full page of notes allowed

```

The `\noteinsert` $\langle text \rangle$ 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

```

1331 \_def\noteinsert #1{\_insert\noteins{%
1332 \noteset
1333 \_vbox to\_ht\_strutbox{\_nobreak \_vskip-\_baselineskip
1334 #1\_unskip\_par \_nobreak \_vskip-\_baselineskip
1335 \_hbox{\_lower\_dp\_strutbox\_vbox{}}
1336 \_penalty0
1337 }}
1338 \_def\noteset{\Heros\cond \_scalemain \_typoscale[800/800] % Heros condensed 80%
1339 \Black \_nobreak
1340 \_widowpenalty=20 \_clubpenalty=20

```

```

1341 \_leftskip=0pt \_rightskip=0pt \_parfillskip=0pt plus1fill
1342 \_parindent=0pt
1343 \_lineskiplimit=-3pt
1344 \_hsize=.5\_hsize \_advance\_hsize by-1em\_relax % two columns
1345 \_everypar{}
1346 }

```

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).

opbible.opm

```

1367 \_addto\_pagecontents{%
1368 \_ifvoid\.noteins \_else
1369 \_vskip\_skip\.noteins \noterule
1370 \_setbox\.noteins=\_vbox{\_penalty0 \_unvbox\.noteins \_vfil}
1371 \_splittopskip=12pt
1372 \_setbox0=\_vsplit\.noteins to0pt % adding \splittopskip to \.noteins
1373 \_def\_Ncols{2}
1374 \_dimen0=.5\_ht\.noteins \_setbox6=\_box\.noteins
1375 \_vskip\_splittopskip
1376 \_balancecolumns
1377 \_fi
1378 \_unless\_ifvoid\.botins \_unvbox\.botins
1379 \_else \_vskip 0pt plus1filll minus8pt \_fi
1380 }
1381 \_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.

opbible.opm

```

1393 \_newinsert\.botins
1394 \_def\.botinsert{\_setbox0=\_vbox\_bgroup}
1395 \_def\.endbot{\_par\_egroup}
1396 \_insert\.botins{\_splittopskip=0pt \_penalty100
1397 \_hrule height0pt \_nobreak\_medskip\_bigskip \_unvbox0
1398 }%
1399 }
1400 \_skip\.botins=\_zoskip % no space added when a topinsert is present
1401 \_count\.botins=1000 % magnification factor (1 to 1)
1402 \_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.

opbible.opm

```

1415 \_def\.putImage #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1416 \_edef\.fullvref{\_gentovref{#1}}%
1417 \_edef\.fullvrefm{\_ea\_renumvref\.fullvref\_relax}%
1418 \_ea\_newaction\_ea{\_fullvrefm}{\_.doImage-#2[#4] (#6){#7}}%
1419 }
1420 \_def\.doImage #1[#2] (#3)#4{% {Title}[label] (params){image-file.pdf}

```

```

1421 \.botinsert
1422 \.botTitle{#1}[#2]%
1423 \_kern3pt \_nobreak
1424 \_hbox{\picw=\hsize #3\inspic{#4}}%
1425 \.endbot
1426 }
1427 \_def\.botTitle#1[#2]{\_hbox{\_captionfont
1428 \_ifx^#2^\_else \.botDest{#1}[#2]\_fi
1429 \_rlap{\Grey \_vrule height1.2em depth.5em width\hsize}\White\_kern12pt #1}%
1430 }
1431 \_picdir={images/}
1432 \_def\.botDest#1[#2]{\_label[#2]\_wlabel{#1}}
1433
1434 \_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 `\balancecolumn` macro.

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

opbible.opm

```

1458 \_newcount\articlenum
1459 \_def\.putArticle #1 #2#3[#4]#5(#6){% chap:verse {Title} [number] (params)
1460 \_edef\.fullvref{\_gentovref{#1}}%
1461 \_edef\.fullvrefm{\_ea\_renumvref\.fullvref\_relax}%
1462 \_ea\_newaction\_ea{\_fullvrefm}{\.doArticle{#2}[#4]#6}%
1463 }
1464 \_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

```

1481 \_def\.doArticle#1[#2] (#3){% {Title}[number] (params)
1482 \_incr\articlenum
1483 \.botinsert
1484 \_def\.botDest##1[##2]{\_trykadedest{a:\_currbook/##2}}
1485 \_parindent=12pt \_iindent=\_parindent
1486 \_setbox0=\_vbox{\_hsize=.458\_hsize \_emergencystretch=1em
1487 \_hbadness=6000 \_baselineskip=\_dimexpr\_baselineskip plus1pt
1488 \_def\Article[##1]{\_endinput}
1489 \_penalty0
1490 \_long\_def\.searcharticle##1\Article[##2]{
1491 \_ea\.searcharticle \_input \articlefile \_relax}
1492 \_splittopskip=12pt
1493 \_setbox1=\_vsplit0 to0pt % adding \splittopskip
1494 \_tmpdim=\_vsize \_advance\_tmpdim by-24pt % \.botTitle height plus above/below skips
1495 \_ifdim 2\_tmpdim > \_ht0 \_tmpnum=1
1496 \_else
1497 \_tmpnum=\_roundexpr{\_bp{\_ht0}/\_bp{1.333\_vsize}+0.999} % number of 2/3 pages
1498 \_fi
1499 \_multiply\_tmpnum by2 % number of columns
1500 \_edef\_Ncols{\_the\_tmpnum}
1501 \_dimen0=\_expr{1/\_Ncols}\_ht0 \_setbox6=\_box0 % height of each two-columns part
1502 \_setbox0=\_vbox{\_balancecolumns}

```

```

1503     \tmpdim=\ht0 \advance\tmpdim by1.2\baselineskip
1504     \setbox0=\vbox{\unvbox0 \global\setbox2=\lastbox}
1505     \setbox0=\hbox{\unhbox2
1506         \fornum 1..\Ncols \do {\unskip \global\setbox1##1=\lastbox}}
1507     \fornumstep -2: \Ncols..1 \do {
1508         \hrule height0pt\kern5pt\nobreak\vfill
1509         \ifnum\Ncols=##1 \botTitle{#1}[#2]\else \botTitle{}[]\fi
1510         \kern3pt \nobreak
1511         \hbox to\hsize{%
1512             \rlap{\LightGrey \vrule height\tmpdim depth6pt width\hsize}%
1513             \kern\parindent
1514             \box1##1\hss\box1\the\numexpr##1-1
1515             \kern\parindent
1516         }
1517         \break
1518     }
1519     \endbot
1520 }
1521 \def\roundexpr#1{\ea\roundexprA\expanded{\expr{#1}}\relax}
1522 \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 $2 * (\hsize * \textit{inner_margin})$. 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

```

1568 \newbox \spanpicbox
1569
1570 \def\insertSpanImage #1#2[#3]#4(#5)#6{%
1571     \checkpicbox
1572     \par \penalty0
1573     \tmpdim=\pagewidth
1574     \advance\tmpdim by-\hoffset
1575     \global\setbox\spanpicbox=\hbox{\picwidth=2\tmpdim \inspic{#6}}
1576     \gdef\startinsertSpanImage {\insertBot {#1}[#3] (#5){\copy\spanpicbox \kern-1.2ex}}
1577     \doinsertSpanImage
1578 }
1579 \def\doinsertSpanImage{%
1580     \ifodd\pageno
1581         \glet\inspicbox=\inspicboxafter
1582     \else

```

```

1583     \_ifdim \_dimexpr \_pagegoal-\_pagetotal > \_dimexpr \_ht\spanpicbox+2em \_relax
1584     \_startinsertSpanImage
1585     \_else
1586         \_glet\inspicbox=\inspicboxafter
1587     \_fi
1588 \_fi
1589 }
1590 \_let\inspicbox=\_useit
1591 \_def\inspicboxafter #1{%
1592     \_ifodd\_pageno
1593         \_startinsertSpanImage
1594         \_glet\inspicbox=\_useit
1595     \_fi
1596 }
1597 \_def \_endoutput{%
1598     \_ifvoid\spanpicbox\_else \_addpicbox\_fi
1599     \_advancepageno
1600     {\_globaldefs=1 \_the\_nextpages \_nextpages={}}%
1601     \_ifnum\_outputpenalty>-20000 \_else\_dosupereject\_fi
1602 }
1603 \_def\addpicbox{\inspicbox{\insertBot{}}{}}{\_moveleft\_pagewidth\_box\spanpicbox\_kern-1.2ex}}
1604
1605 \_def\checkpicbox{%
1606     \_ifvoid\spanpicbox\_else \_errmessage{Two span Image/Text at single place not allowed}\_fi
1607 }

```

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

opbible.opm

```

1617 \_long\_def\insertSpanText #1#2[#3]#4(#5)#6{%
1618     \_checkpicbox
1619     \_par \_penalty0
1620     \_tmpdim=\_pagewidth
1621     \_advance\_tmpdim by-\_hoffset
1622     \_setbox0=\_hbox to2\_tmpdim{\_hss\_vbox{\_hszize=2\_tmpdim
1623         \_leftskip=0pt \_rightskip=0pt \_relax \_kern3pt #6}\_hss}
1624     \_global\_setbox\spanpicbox=
1625         \_hbox{\_rlap{\_White \_vrule width\_wd0 height\_ht0 depth\_dp0}\_box0}
1626     \_global\_ht\spanpicbox=\_dimexpr\_ht\spanpicbox-3pt\_relax
1627     \_gdef\startinsertSpanImage {\insertBot {#1} [#3] (#5) {\_copy\spanpicbox \_kern-1.2ex}}
1628     \_doinsertSpanImage
1629 }
1630 \_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 $c + 2$, $c + 3$.

opbible.opm

```

1646 \_newcount\spantextnum
1647 \_def\putSpanImage #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1648     \_edef\fullvref{\_gentovref{#1}}%
1649     \_edef\fullvrefm{\_ea\_renumvref\fullvref\_relax}%
1650     \_ea\_newaction\_ea{\fullvrefm}{\insertSpanImage{#2} [#4] (#6) {#7}}%
1651 }
1652 \_long\_def\putSpanText #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1653     \_edef\fullvref{\_gentovref{#1}}%
1654     \_edef\fullvrefm{\_ea\_renumvref\fullvref\_relax}%
1655     \_incr\spantextnum
1656     \_global\_sdef{spant!\the\spantextnum}{#7}%
1657     \_ea\putSpanTextA
1658     \_expanded{\fullvrefm}\_ea\_csname spant!\the\spantextnum\_endcsname {#2} [#4] (#6)%
1659 }

```



```

1660 \_def\putSpanTextA #1#2#3[#4] (#5){\newaction{#1}{\insertSpanText{#3}[#4] (#5){#2}}}
1661
1662 \_nspublic \putSpanImage \putSpanText ;

```

15 Inserting citations to the page

`\putCite` $\langle gen-vref \rangle$ $\{ \langle text \rangle \}$ creates a citation $\langle text \rangle$ inserted to the top of the page where the verse $\langle gen-vref \rangle$ is. We register a new action by `\newaction` $\{ \langle full-vref \rangle \} \{ \dotopCite \{ \langle text \rangle \} \}$.

opbible.opm

```

1674 \_def\putCite #1 #2{% chap:verse {text}
1675   \_edef\fullvref{\gentovref{#1}}%
1676   \_edef\fullvrefm{\_ea\renumvref\fullvref\_relax}%
1677   \_ea\newaction\_ea{\fullvrefm}{\dotopCite{#2}}%
1678 }
1679 \_nspublic \putCite ;

```

`\dotopCite` $\{ \langle text \rangle \}$ 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 saw the page position using `\ewref` to the .ref file as `\sxdef{ct! \langle citenum \rangle}{\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.

opbible.opm

```

1691 \_newcount\citenum
1692 \_def\dotopCite #1{%
1693   \topinsertnopar
1694   \_typosize[12/16]\_bi
1695   \_incr\citenum
1696   \_ifodd \_trys{ct!\_the\citenum}{0}\_relax
1697     \_leftskip=.3\_hsize plusfil \_parfillskip=0pt
1698     \_noindent
1699     \_rlap{\_hskip\_hsize \_kern-\_leftskip \_copy\lqqbox}\_hfill
1700   \_else
1701     \_let\quotedby=\_quotedbyright
1702     \_rightskip=.3\_hsize plus 1fil
1703     \_noindent \_llap{\_copy\lqqbox}%
1704   \_fi
1705   {\_printCite{#1}\_unskip}\_par
1706   \_ewref\_sxdef{ct!\_the\citenum}{\_string\mypage}}%
1707 % \_vskip-.3\_baselineskip
1708 \_endinsert
1709 }
1710 \_def\printCite#1{\_pdfliteral{2 Tr .15 w .9 g}#1\_pdfliteral{0 Tr 0 w 0 g}}
1711 \_def\printCite#1{\_Grey#1}}
1712
1713 \_def\topinsertnopar{\_umidfalse \_upagefalse \_begingroup\_setbox0=\_vbox\_bgroup\_resetattrs}

```

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` $\{ \langle author \rangle \}$ puts the author of the quotation to the next line. The macro `\quotedbyright` (which is used at left pages) prints the $\langle author \rangle$ at the last line if there is sufficient space.

opbible.opm

```

1723 \_newbox\lqqbox
1724 \_newbox\rqqbox
1725 \_setbox\lqqbox=\_hbox{\_lower3pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed,,}}
1726 \_setbox\rqqbox=\_hbox{\_kern2pt\_lower38pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed" }}
1727 \_ht\lqqbox=0pt \_dp\lqqbox=0pt
1728 \_ht\rqqbox=0pt \_dp\rqqbox=0pt
1729 \_addto\enquotes{\_setbox0=\_box\lqqbox \_setbox\lqqbox=\_box\rqqbox \_setbox\rqqbox=\_box0 }
1730
1731 \_def\quotedby{\_par}
1732 \_def\quotedbyright#1{%
1733   \_unskip\_nobreak\_hfill\_penalty0\_hskip2em
1734   \_null\_nobreak\_hskip\_iindent\_hbox{#1}}

```

The following macros `\Cite`, `\insertCite` and `\swapCites` are used for insertion of citations to the two-column printed articles. The `\Cite` $\langle label \rangle \{ \langle text \rangle \}$ simply saves the $\langle text \rangle$ to the macro `\c!` $\langle article-num \rangle ! \langle label \rangle$. The `\insertCite` $\langle label \rangle \langle left-or-right \rangle$ inserts the citation declared by `\Cite`

$\langle label \rangle$ to the text using $\backslash\text{vadjust}$. The variant $\backslash\text{left}$ and $\backslash\text{right}$ is processed or ignored. This depends on the parity of the current page, which is restored from .ref file and saved to the macro $\backslash\text{cp}\{ \langle article\text{-num} \rangle \} \{ \langle label \rangle \}$.

opbible.opm

```

1748 \_def\Cite #1#2{\_sdef{c!\_the\articlenum!#1}-#2}}
1749 \_def\insertCite #1#2{\_def\citelabel{#1}%
1750   \_ifx\_left#2\insertCiteleft
1751   \_else \_ifx#2\_right\insertCiteright\_else
1752     \_errmessage{\_noexpand\insertCite#1: \_noexpand\left or \_noexpand\right expected}%
1753   \_fi\_fi
1754 }
1755 \_def\insertCiteleft {%
1756   \_ifnum\citepg=1
1757     \_printwarn{\_noexpand\insertCite\citelabel: \_noexpand\swapCites activated}\_fi
1758   \_ifodd \_numexpr\_trycs{cp!\_the\articlenum!\_citelabel}\_citelabel\citepg\_relax
1759   \_else \_insertCitelr \_left \_fi
1760 }
1761 \_def\insertCiteright{%
1762   \_ifodd \_numexpr\_trycs{cp!\_the\articlenum!\_citelabel}\_citelabel\citepg\_relax
1763   \_insertCitelr \_right \_fi
1764 }
1765 \_def\insertCitelr#1{\_unskip\_vadjust{\_vbox{%
1766   \_ewref\_sxdef{cp!\_the\articlenum!\_citelabel}\_string\mypage}}%
1767   \_vskip6pt
1768   \_advance\_hsize by\_parindent
1769   \_typosize[12/16]\_bi\Grey
1770   \_ifx#1\_left
1771     \_def\quotedby{\_par\_hfill}
1772     \_rightskip=\_parindent plus1fil \_leftskip=0pt
1773     \_setbox0\_vbox{%
1774       \_medskip \_noindent
1775       \_llap{\_copy\lqqbox}\_ignorespaces
1776       \_printCite{\_cs{c!\_the\articlenum!\_citelabel}}\_medskip}%
1777     \_hbox{\_kern-\_parindent\_rlap{\_White
1778       \_vrule height\_ht0 width\_hsize}\_box0}%
1779   \_else
1780     \_leftskip=\_parindent plus1fil
1781     \_parfillskip=0pt
1782     \_setbox0\_vbox{%
1783       \_medskip \_noindent
1784       \_rlap{\_hskip\_hsize\_kern-\_parindent\_copy\lqqbox}\_hfill
1785       \_ignorespaces \_printCite{\_cs{c!\_the\articlenum!\_citelabel}}\_medskip}%
1786     \_rlap{\_rlap{\_White \_vrule height\_ht0 width\_hsize}\_box0}%
1787   \_fi
1788   \_vskip6pt
1789 }}}
1790 \_def\swapCites{\_def\citepg{1}}
1791 \_def\citepg{0}
1792
1793 \_nspublic \Cite \insertCite ;

```

Insertions into the intro text

opbible.opm

```

1801 %% TBN page 236
1802
1803 \_newcount\shapenum
1804 \_newdimen\ii \_newdimen\w
1805 \_def\oblong #1 od #2 odsadit #3 {\_par \_ii=#1 \_w=\_hsize
1806   \_ifdim\_ii>\_zo \_advance\w by-\_ii
1807   \_else \_advance\w by\_ii \_ii=\_zo \_fi
1808   \_shapenum=1 \_tmpnum=0 \_def\shapelist{}
1809   \_loop \_ifnum\shapenum<#2 \_edef\shapelist{\_shapelist\_zo\_hsize}%
1810     \_advance\shapenum by1 \_repeat
1811   \_loop \_edef\shapelist{\_shapelist\_ii\w}%
1812     \_advance\_tmpnum by1 \_ifnum\_tmpnum<#3 \_repeat
1813   \_advance\shapenum by#3 \_edef\shapelist{\_shapelist\_zo\_hsize}
1814   \_doshape}
1815 \_def\shapelist{\_parshape \_shapenum \_shapelist}
1816 \_newcount\globpar

```

```

1817 \_ifx\_partokenset \_undefined \_def\.partoken{\par} \_else \_def\.partoken{\par} \_fi
1818 \_def\.doshape{\_global\_.globpar=0 \_ea\_def\.partoken{\_ifhmode\_.shapepar\_fi}}
1819 \_def\.shapepar{\_prevgraf=\_globpar \_parshape\_.shapenum\_.shapelist
1820 \_endgraf \_global\_.globpar=\_prevgraf
1821 \_ifnum \_prevgraf>\_.shapenum \_ea\_let\.partoken=\_endgraf \_fi
1822 }
1823
1824 \_def\.Citehereleft #1 (#2) #3{{
1825 \_par
1826 \_def\quotedby{\_par\_hfill}
1827 \_rightskip=\_parindent plus1fil \_leftskip=0pt
1828 \_setbox0\_vbox{{%
1829 \_typosize[12/16]\_bi\Grey
1830 \_hsize=.5\_hsize
1831 \_medskip \_noindent
1832 \_llap{\_copy\_.lqqbox}\_ignorespaces
1833 \_printCite{#3}\_medskip}}%
1834 \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1835 \_xdef\.lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1836 \_nointerlineskip\_vbox toOpt{\_kern#1\_baselineskip #2
1837 \_hbox{\_rlap{White
1838 \_kern-3mm\_vrule height\_ht0 width.5\_hsize}\_box0}}%
1839 \_vss}}
1840 \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1841 \_oblong {.5\_tmpdim} od #1 odsadit {\.lines}
1842 }
1843 \_def\.Citehereright #1 (#2) #3{{
1844 \_par
1845 \_def\quotedby{\_par\_parfillskip=0pt \_hfill}
1846 \_leftskip=\_parindent plus1fill \_rightskip=0pt
1847 \_setbox0\_vbox{{%
1848 \_typosize[12/16]\_bi\Grey
1849 \_hsize=.5\_hsize
1850 \_vskip\_medskipamount \_rlap{\_kern\_hsize\_copy\_.rqqbox}\_vskip-\_medskipamount
1851 \_printCite{\_noindent\_ignorespaces#3}\_medskip}}%
1852 \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1853 \_xdef\.lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1854 \_nointerlineskip\_vbox toOpt{\_kern#1\_baselineskip #2
1855 \_hbox to\_hsize{\_hss
1856 \_llap{White \_vrule height\_ht0 width.5\_hsize \_kern-3mm}%
1857 \_llap{\_box0}}
1858 \_vss}}
1859 \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1860 \_oblong {-.5\_tmpdim} od #1 odsadit {\.lines}
1861 }
1862
1863 \_def\.Citehere{\_par \_ifodd\_pageno \_ea\.Citehereright \_else \_ea\.Citehereleft \_fi}
1864
1865 \_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.

opibble.opm

```

1881 \_def\.insertBot #1#2[#3]#4(#5)#6{% {Title} [label] (params) {data}
1882 \_botinsert
1883 \_leftskip=0pt \_rightskip=0pt \_relax
1884 \_botTitle{#1}[#3]%
1885 \_kern3pt \_nobreak
1886 \_vbox{\_picwidth=\_hsize #5 #6}%
1887 \_endbot
1888 }
1889 \_def\.putBot #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1890 \_edef\.fullvref{\_gentovref{#1}}%

```

```

1891 \_edef\fullvrefm{\_ea\renumvref\fullvref\_relax}%
1892 \_ea\newaction\_ea{\fullvrefm}{\insertBot{#2}[#4](#6){#7}}%
1893 }
1894 \_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

```

1903 \_def\.printintro{%
1904 \.begblock
1905 \_dest[i:\currbook/]
1906 \_chaptit{\_mtext{intro}}%
1907 \_input{\introfile}
1908 \.endblock
1909 }

```

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

```

1917 \_newcount\blocklevel % nesting level of blocks
1918 \_def\.begblock{\_par\_bgroup
1919 \_advance\blocklevel by1 \_advance\_leftskip by\_iindent \_rightskip=\_leftskip
1920 \_medskip
1921 \_pdfsavepos \_ea\_wref\_ea\Xblock\_ea{\_ea{\_the\blocklevel}B{\_the\_pdflastypos}}
1922 \_nobreak \_medskip
1923 }
1924 \_def\.endblock{\_par\_nobreak\_medskip
1925 \_pdfsavepos \_ea\_wref\_ea\Xblock\_ea{\_ea{\_the\blocklevel}E{\_the\_pdflastypos}}
1926 \_medskip \_egroup
1927 }
1928 \_refdecl{%
1929 \_def\Xblock#1#2#3{\_ifnum#1=1 \_edef\tmp{frm:\_ea\_ignoresecond\_currpage}^^J
1930 \_unless\_ifcname \_tmp \_endcname \_sxdef{\_tmp}{\_fi}^^J
1931 \_sxdef{\_tmp}{\_cs{\_tmp}#2#3}\_fi}
1932 }
1933 \_newdimen\frtop \_newdimen\frbottom % positions of top and bottom text on the pages
1934 \_def\frcolor{.93 g } % light grey -- color of blocks.
1935 \_pgbackground={%
1936 \_slet{\_opb\_tmp}{frm:\_the\_gpageno}
1937 \_ifx\_tmp\_undefined \_def\tmp{\_fi}
1938 \_frtop=\_dimexpr \_pdfpageheight-\_voffset+\_smallskipamount\_relax
1939 \_frbottom=\_dimexpr \_pdfpageheight-\_voffset-\_vsize-\_medskipamount\_relax
1940 \_ifx\_frnext y \_edef\tmp{B{\_number\frtop}\_tmp}\_global\_let\_frnext n\_fi
1941 \_ea\printframes \_tmp B{0}E{\_number\frbottom}
1942 \_ifx\frameslist\_empty \_else
1943 \_pdfliteral{q \_frcolor 1 0 0 1 0 \_bp{-\_pdfpageheight} cm \_frameslist Q}\_fi
1944 }
1945 \_def\printframes B#1#2E#3{\_ifnum#1=0 \_else
1946 \_printframe {\_hoffset}{#3sp}{\_xhsize}{\_ifnum#1=-1 \_number\frtop\_else#1\_fi sp-#3sp}
1947 \_ifx^#2\_else \_global\_let\_frnext=y \_let\printframes=\_relax \_fi
1948 \_ea\printframes\_fi}
1949 }
1950 \_def\frameslist{}
1951 \_def\printframe #1#2#3#4{\_edef\frameslist{\_frameslist
1952 \_bp{#1} \_bp{#2} \_bp{#3} \_bp{#4} re f }%
1953 }

```

Insertions objects over pictures (maps)

`\putstext` $\langle x\text{-pos} \rangle \langle y\text{-pos} \rangle \{ \langle \textit{text} \rangle \}$ 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` $\{ \langle \textit{text} \rangle \}$ creates an `\hbox` $\{ \langle \textit{text} \rangle \}$ 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

```

1974 \_def\putstext{\_ea\_ea\_ea\putstextA\_scantwodimens}
1975 \_def\putstextA#1#2#3{%

```

```

1976 \_setbox0=\_hbox{\_shadowedtext{#3}}%
1977 \_dimen1=#1sp \_dimen2=#2sp \_puttextB
1978 }
1979 \_def\_shadowedtext#1{%
1980 \_insertwhiteshadowresources
1981 \_setbox0=\_hbox{#1}%
1982 \_hbox{\_tmpdim=\_ht0 \_advance\_tmpdim by\_dp0
1983 \_lower\_dp0\_hbox{%
1984 \_pdfliteral{q /trans gs 1 g
1985 \_for num 1..10\_do{\_oval{\_bp{\_wd0}}{\_bp{\_tmpdim}}{2+##1/2} f } Q}}%
1986 \_box0}%
1987 }
1988 \_def\_insertwhiteshadowresources{%
1989 \_addextgstate{trans}{<</ca \shadowparameter>>}%
1990 \_glet\_insertwhiteshadowresources=\_relax
1991 }
1992 \def\shadowparameter{.1} % default value of "transparency"
1993
1994 \_nspublic \_puttext \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

```

2003 \_def\_c[#1/#2]#3{% text podel krivky: \c[init-rotace/repetice]{text}
2004 \_pdfsave\_pdfrotate{#1}\_rlap{\_let\_printwarn=\_ignoreit
2005 \_edef\_tmpb{#3}\_replstring\_tmpb{ }{{ }}\_def\_tmpa{#2}%
2006 \_ea\_foreach\_tmpb\_do{##1.\tmpa}}\_pdfrestore \_kern10mm
2007 }
2008 \_let\_c=\_undefined
2009 \_nspublic \_c ;

```

`\town dimen dimen` puts a circle with given `\townparams` to the given place *dimen* *dimen*. It works like `\puttext dimen dimen {circle}`.

opbible.opm

```

2017 \_def\_townparams{ % default parameters of the circle:
2018 \_hhkern=.8pt % diameter of the disc
2019 \_lwidth=.5pt % tickness of the outline
2020 \_fcolor=\Red % color of the inner disc
2021 \_lcolor=\Black % color of the outline
2022 }
2023 \_def\_town {\_ea\_ea\_ea.\townA\_scantwodimens}
2024 \_def\_townA #1#2{\_setbox0=\_hbox{\_incircle[\_hhkern=0pt \_vbkern=0pt \townparams]{}%
2025 \_dimen1=#1sp \_dimen2=#2sp \_puttextB
2026 }
2027 \_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

```

2050 \_def\_easylist{\_ade*{\_countlist}}
2051 \_def\_aast{\_countlist}
2052 \_def\_countlist{\_tmpnum=1 \_countlistA}
2053 \_def\_countlistA{\_futurelet\_next\_countlistB}

```

```

2054 \_def\countlistB{\_ifx\next\ast \ea\countlistC\else \ea\countlistD \fi}
2055 \_def\countlistC#1{\_incr\tmpnum \countlistA}
2056 \_def\countlistD{%
2057   \_ifnum\tmpnum>\_ilevel \fornum \ilevel..\tmpnum-1 \do{\begitemseasylist}\else
2058   \_ifnum\tmpnum<\_ilevel \fornum \tmpnum..\ilevel-1 \do{\enditemseasylist}\fi\fi
2059   \startitem}
2060
2061 \_def\qq#1{\_bf#1\trycs{Level:\_the\ilevel}{}}\space\aftergroup\qqA}
2062 \_def\qqA{\_sdef{Level:\_the\ilevel}{\_rlap'}}}
2063 \_def\ChiasmNumbering{\ea\qq \Uchar \numexpr `A-1+\ilevelrelax\space} % A, B, C, D, etc.
2064 \_sdef{item:q}{}%for chiasms with no leading alphabet letters
2065 \_sdef{item:Q}{\ChiasmNumbering}
2066 \_def\beginChiasm{\begitemseasylist \style Q \let\_defaultitem=\printitem}
2067 \_def\endChiasm{\fornum 1..\ilevel \do{\enditemseasylist}}
2068
2069 \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.

```

2081 \_newdimen\colsep
2082 \colsep=10pt
2083
2084 \_def\Outline{
2085   \medskip
2086   % \filbreak
2087   \chaptit{\mtext{outline}}%
2088   \everylist={\_ifcase\ilevel \or \style I \or \style A \or \style n \fi}
2089   \_sdef{item:A}{\_strut\_uppercase\ea{\_athe\itemnum}. }
2090   \_sdef{item:I}{\_strut\_uppercase\ea{\romannumeral\itemnum}. }
2091   \_hsize=.5\hsize \advance\_hsize by-\colsep
2092   \emergencystretch=40pt
2093   \leftskip=0pt \rightskip=0pt
2094 }
2095 \_def\rightnote#1{\_par
2096   \setbox0=\hbox{\_kern\hsize \kern\colsep
2097     \vtop{\_leftskip=0pt \kern0pt\_noindent\_strut\_it#1}}
2098   \ht0=0pt \dp0=0pt \box0 \nointerlineskip
2099 }
2100 \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).

```

2113 \_def\l{\_baselineskip}
2114 \_newcount\timeline \timeline=100 % default
2115 \_newdimen\tlwidth \tlwidth=10cm % default
2116 \_def\timelinewidth{\_afterassignment\timelinewidthA\tlwidth}
2117 \_def\timelinewidthA{\_par\_hbox to\tlwidth{}}
2118
2119 \let\l=\_undefined
2120 \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.

```

2133 \_def\arrowtext #1..#2(#3)#4{%
2134 \_puttext \.pos{#1}Opt
2135 {\_lower.745ex\_hbox to\_dimexpr\.pos{#2}-\.pos{#1}{#3\.\Larrow{ #4 }\.\Rarrow}}
2136 }
2137 \_def\.\Larrow{${\leftarrow$\_kern-.8em\_leaders\_vrule height.65ex depth-.42ex\_hfil}
2138 \_def\.\Rarrow{\_leaders\_vrule height.65ex depth-.42ex\_hfil\_kern-.8em$\rightarrow$}
2139 \_def\.\rule{\_leaders\_vrule height.12ex depth.12ex\_hfil}
2140 \_def\.\pos#1{\_expr{#1/\_the\.\timeline}\.\tlwidth}
2141
2142 \_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.

```

2155 \_def\.\tlput #1 #2 #3(#4)#5{%
2156 \_let\.\Lhss=\_hss \_let\.\Rhss=\_hss
2157 \_ifx#3\_rlap\_relax \_let\.\Lhss=\_relax \_let\.\Rhss=\_hss \_fi
2158 \_ifx#3\_llap\_relax \_let\.\Lhss=\_hss \_let\.\Rhss=\_relax \_fi
2159 \_puttext \.pos{#2}Opt {\_hbox toOpt{\_Lhss #4\.\tltext#1{#5}\.\Rhss}}
2160 }
2161 \_def\.\tltext#1#2{\_ifx#1a\_vbox\_else
2162 \_vtop\_fi{\_kernOpt\_halign{\_Lhss#\.\Rhss\_cr\_strut#2\_crr}}}%
2163 }
2164 \_nspublic \tlput ;

```

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

\tlines*<{data/separated/by}</i> 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.*

```

2174 \_def\.\tline #1..#2 {%
2175 \_puttext \.pos{#1}Opt {\_hbox to \_dimexpr\.pos{#2}-\.pos{#1}{\.\rule}}
2176 }
2177 \_def\.\tlines#1{\_puttext OptOpt{\_hbox{\_foreach #1|\_do##1|{\_vrul\_hskip\.pos{0##1}}}}
2178 \_def\.\vrul{\_def\.\vrul{\_kern-.12ex\_vrule height.7\l depth.7\l width.24ex \_kern-.12ex}}
2179
2180 \_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.

```

2194 \_def\.\normalchapnumbers{
2195 \_margins/2 a4 (25,25,20,20)mm
2196 \_lrmargin=0pt
2197 \_setbox0=\_box\.\lqqbox \_setbox0=\_box\.\rqqbox
2198 \_def\.\printbeforefirst{%
2199 \_nobreak\_medskip
2200 \_trychapnote
2201 \_hangindent=\_parindent \_hangafter=-2
2202 \_noindent \_llap{\_vbox toOpt
2203 {\_kern-8pt\_hbox{\_setfontsize{at23pt}\_bf\Red\_the\.\chapnum\_kern5pt}\_vss}}%
2204 }
2205 }
2206 \_nspublic \normalchapnumbers ;

```

20 Checking syntax

```

2214 \_def\.\checksyntax#1 {%
2215 \_let\processbooks=\_relax
2216 \_ifx\_relax#1\_relax \_else

```

```

2217 \begingroup
2218 \_the\syntaxmacros
2219 \_wterm{^^J** checking file: #1 **^^J}
2220 \_input{#1}
2221 \_vfil\_break
2222 \_endgroup
2223 \_ea\checksyntax \_fi
2224 }
2225
2226 \_newtoks\syntaxmacros
2227 {\_catcode`<=13
2228 \_global\syntaxmacros={
2229 \_def<#1>{\_bgroup
2230 \_message{checking \_unexpanded{<#1>}}%
2231 \_ifx\_relax#1\_relax \_errmessage{empty link}\.nobref\_else \_afterfi{\.checkbref#1>\.bref#1>}\_fi
2232 \_glet\.linkpre=\.linkpre \_glet\.linkfspec=\.linkfspec
2233 \_egroup
2234 }
2235 \_def\.checkbref#1#2>{%
2236 \_isinlist{.#1#2}{<}\_iftrue \_errmessage{duplicated \_string<}\.nobref\_else
2237 \_ifx"#1\.checkbrefQ #1#2>\_else \.checkbrefD #1#2>\_fi\_fi
2238 }
2239 \_def\.checkbrefQ "#1"#2#3>{\.checkbrefD #2#3>}
2240 \_def\.checkbrefD #1>{%
2241 \_isinlist{.#1}{ }\_iftrue\.checkbrefS#1>\_else\.checkbrefN#1>\_fi
2242 }
2243 \_def\.checkbrefS #1 #2>{\.checkbrefN#2>}
2244 \_def\.checkbrefN #1#2>{%
2245 \_def\.tmpb{#1}
2246 \_ifx\.tmpb\_empty \_errmessage{missing link data}\.nobref\_else
2247 \_replstring\.tmpb{:}{ }\_replstring\.tmpb{-}{ }\_replstring\.tmpb{ }{ }%
2248 \_replstring\.tmpb{a}{ }\_replstring\.tmpb{b}{ }\_replstring\.tmpb{c}{ }%
2249 \_setbox0=\_hbox{\_tmpnum=0\.tmpb\_relax}%
2250 \_ifdim\_wd0>opt \_errmessage{nonnumeric link data}\.nobref\_fi
2251 \_fi
2252 }
2253 \_def\.nobref{\_def\.bref##1>{\Red\_string<##1>}}
2254 \_def\.currbook{}
2255 \_def\.prelinkB{BK}
2256 \_def\.prelinkC{BK}
2257 \_def\.prelinkV{0}
2258 \_def\nochapbooks{BK}
2259 \_let\<=<
2260
2261 \_def\x/#1/{\_def\.tmpb{#1}%
2262 \_isinlist\.tmpb\x\_iftrue \.badx
2263 \_else \_isinlist\.tmp<\_iftrue \.badx
2264 \_else \_isinlist\.tmp\enditems\_iftrue \.badx \_else \.x/#1/\_fi\_fi\_fi
2265 }
2266 \_def\.badx{\_errmessage{unclosed \_string\x/.../}}
2267
2268 \_def\Article[#1]{}
2269 \_def\Cite #1 {\_par\_noindent{\_bf Cite: }}
2270 \_def\insertCite #1#2{}
2271
2272 \_def\putArticle #1 #2[#3]#4(#5){}
2273 \_def\putCite #1:#2 {\_par\_noindent{\_bf Cite: }}
2274 \_def\putBot #1 #2[#3]#4(#5){\_vbox}
2275
2276 \_def\c[#1/#2]#3{#3}
2277
2278 \_long\_ea\_def\_csname Note\_endcsname #1 #2#3%
2279
2280 {\_par \_let\.nextww\_undefined \_noindent{\_bf Note #1:} #3\_par}
2281 }}
2282 \_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.

opbible.opm

```
2297 \_newwrite\.outfile
2298 \_def\.filegen #1 {\_par
2299   \_begingroup \_addto\genbooks{ }\_def\.filename{#1}%
2300   \_setverb \_endlinechar=``^J \.filegenA
2301 }
2302 \_ea\_def \_ea\.filegenA \_expanded{#1^^J\_csstring\\endfile#2^^J}{%
2303   \_def\.filecontent{#1}%
2304   \_ea\_foreach\genbooks \_do ##1 {%
2305     \_bgroup
2306     \_ifx^##1^\_else
2307       \_replstring\.filename{@@}{##1}%
2308       \_isfile{\.filename}\_iftrue \_opwarning{file "\.filename" exists already}%
2309     \_else
2310       \_wterm{creating file: \.filename}%
2311       \_immediate\_openout\.outfile={\.filename}%
2312       \_replstring\.filecontent{@@}{\.btitle{##1}}%
2313       \_replstring\.filecontent{@@}{##1}%
2314       \_immediate\_write\.outfile{\.filecontent}\_immediate\_closeout\.outfile
2315     \_fi\_fi
2316     \_egroup
2317   }%
2318   \_endgroup
2319 }
2320 \_def\.btitle#1{\_ifcsname fb!#1\_endcsname \_trycs{btitle!\_cs{fb!#1}}{#1}%
2321   \_else \_trycs{btitle!#1}{#1}\_fi
2322 }
2323 \_nspublic \filegen ;
```

22 Other macros

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

opbible.opm

```
2333
2334 \_def\.quotationmarks#1#2{%
2335   \_cnvtext{"}{\_doquotationmark}%
2336   \_def\.doquotationmark {\_futurelet\_next\_doquotationmarkA}%
2337   \_def\.doquotationmarkA {%
2338     \_let\.doquotationmarkB=#1\_relax
2339     \_ea\_ifx\_space\_next \_let\.doquotationmarkB=#2\_fi
2340     \_ifx\_space\_next \_let\.doquotationmarkB=#2\_fi
2341     \_ifx\_endgraf\_next \_let\.doquotationmarkB=#2\_fi
2342     \_ifx\_empty\_next \_let\.doquotationmarkB=#2\_fi
2343     \_ifx.\_next \_let\.doquotationmarkB=#2\_fi
2344     \_ifx,\_next \_let\.doquotationmarkB=#2\_fi
2345     \_doquotationmarkB}%
2346 }
2347 \_nspublic \quotationmarks ;
2348
2349 \_def\.chaptit#1{\_line{\_hss\.chapfont\Red#1\_hss}
2350   \_nobreak
2351 }
2352 \_def\.schaptit#1{\_bigskip\.chaptit{#1}\_nobreak\_medskip}
2353
2354 \_def\.subtit#1{\_par
2355   \_ifnum\.currversenum=1 \_else \_medskip\_fi
2356   \_line{\_indent\.subtitfont #1\_hss}\_nobreak
2357   \_ifnum\.currversenum=1 \_vskip-\_medskipamount\_fi
2358   \_smallskip
2359 }
2360 \_def\.subtitfont {\Red\_it}
2361
```

```

2362 \_nspublic \chaptit \schaptit \subtit ;
2363
2364 \_sdef{\_mt:intro:en}{Introduction} \_sdef{\_mt:outline:en}{Outline}
2365 \_sdef{\_mt:intro:cs}{Úvod} \_sdef{\_mt:outline:cs}{Osnova}
2366
2367 \_def\dopsat{{\Red !!! DOPSAT !!! }}
2368
2369 \_def\.bibleinput#1 {\_bgroup
2370 \_catcode`##=13 \_bgroup\_lccode`~=`## \_lowercase{\_egroup\_let~}=\_processline
2371 \_input{#1}%
2372 \_egroup
2373 }
2374 \_let\FormattedBook=\_ignoreit % for backward compatibility
2375 \_let\CommentedBook=\_ignoreit % for backward compatibility

```

23 Setting active character and \outer macros

Active character < used for references.

opbible.opm

```

2384 \_outer\_def\Note {\_Note}
2385 \_outer\_def\ww {\_ww}
2386 \_outer\_def\ChapterPre {\_ChapterPre}
2387 \_outer\_def\ChapterPost {\_ChapterPost}
2388 \_outer\_def\BookTilte {\_BookTitle}
2389
2390 \_def\_afterload{\_adef<{\_bref}}
2391 \_afterload
2392
2393 \_endnamespace

```

24 Index

\.AddNote 6	\.buff 5, 8–11	\.filename 30
\.addpre 9	\c 26	\.finalwork 3–4
\alist! 5, 10	\centeringmode 9	\fmtfile 3
\amark 3	\.chapafter 11	\fmtfont 10
\arrowtext 27	\.chapbefore 11	\fmtins 5, 9
\Article 19	\.checknochapbooks 4	\fmtkeep 5, 10
\.begblock 25	\Cite 22	\.fmtpoetA 9
\begcenter 9	\.Cnotetext 8	\.fmtpoetB 9
\begChiasm 26	\cnvtext 11	\.fmtpoetC 9
\bex! 3–4	\.createlink 14	\fmtpoetry 9, 11
\.bibleinput 3	\.currbook 3, 6, 10	\fmtpre 9, 11
\bibname 4	\.currchapnum 10	\.fmtprebuf 11
\bmark 3–4	\.currverse 10	\.fmtprebuff 9
\BookException 3–4	\.currversenum 10–11	\fmtprepoet 9
\BookPost 3–4	\.currversetext 10	\fmtrepl 10
\BookPre 3–4	\.doArticle 19	\ftmadd 9
\BookTitle 4, 30	\.doCNote 8	\.fullvref 6
\.botins 18	\.doImage 18	\.fullvrefm 6
\.botinsert 18–19	\.doNote 6–8	\genbooks 30
\.botTitle 18	\.dotopCite 22	\.gentovref 6
\bpo! 3–4	\.endblock 25	\.hboxorllap 9, 11
\bpr! 3–4	\.endbot 18–19	\ind 9, 11
\.bref 12	\endcenter 9	\.iniheadline 4
\.brefBookChapter 4	\endChiasm 26	\insertBot 24
\.brefL 13	\.ensuredest 14	\insertCite 22
\.btit 3	\f! 4	\insertSpanImage 20–21
\btit! 4	\.filecontent 30	\insertSpanText 21
\.btitle 30	\filegen 30	\introfile 3, 25

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