

OP-Bible – Technical Documentation

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

`op-bible.opm`

```
3 \_codedecl \processbooks {OpBible: macros for creating annotated Bible}
4
5 \_message{This is OP-Bible, version <0.15 Dec 2022>}
```

1 Preparatory work

Loading packages.

`op-bible.opm`

```
14 \_load[vlna] % single-letter prepositions and splitting hyphen managed specially in Czech
15 \_load[mte] % micro typographical extensions
16
17 \_namespace{opb}
```

Basic settings of \TeX parameters.

`op-bible.opm`

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

Fonts.

`op-bible.opm`

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

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⟩}`.

`op-bible.opm`

```
67 \_let\printwarn=\opwarning
68 \_def \.sedef #1{\_ea\_edef \_csname#1\_endcsname}
69 \_long\_def\myaddto#1#2{\_ifcsname#1\_endcsname
70 \_gobal\_ea\_addto\_csname#1\_endcsname#2\_else \_global\sdef{#1}{#2}\_fi}
```

We prepare expandable if-macros:

`\.isspacein` $\langle text \rangle$ `_iftrue` is true if $\langle text \rangle$ includes a space.
`\.iscolonin` $\langle text \rangle$ `_iftrue` is true if $\langle text \rangle$ includes a colon.
`\.isdivisin` $\langle text \rangle$ `_iftrue` is true if $\langle text \rangle$ includes a divis.

op-bible.opm

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

2 The main loop over Bible books

The `\processbooks` macro does two loops over all marks in `\printedbooks`. The macro `\printedbooks` is a list of $\langle a\text{-marks} \rangle$ 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 $\langle space \rangle \{ \}$ at the end of the expanded `\printedbooks`. The first loop body sets `\pbook!` $\langle a\text{-mark} \rangle$ 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 formatin of the introduction text is done by the `\.printintro` macro.
- Inputs format definition file if it exists. Information is saved to the \TeX memory.
- Inputs notes file if it exists. The notes are saved to the \TeX memory.
- Calls `\bpr!\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.

op-bible.opm

```
121 \_def\.processbooks {\_par
122   \_ifx\tmark\_undefined \_def\tmark{none}\_fi
123   \.checknochapbooks
124   \_useit{\_ea\.processbooksA \printedbooks} {}
125   \_useit{\_ea\.processbooksB \printedbooks} {}
126 }
127 \_def\.processbooksA #1 {%
128   \_if\_relax#1\_relax \_else \_sxddef{pbook!#1}\}\_ea\.processbooksA \_fi
129 }
130 \_def\.processbooksB #1 {%
131   \_if\_relax#1\_relax \_else
132     \_edef\amark{#1}
133     \_edef\bmark{\_cs{f!#1}}
134     \_edef\.btit{\_cs{btit!#1}}
135     \_begingroup
136       \_edef\.currbook{#1}
137       \.newbook{#1}
138       \_wterm{** \_cs{btit!#1} {#1} (\string\tmark: \tmark) **}
139       \_cs{bex!#1}
140       \_isfile{\introfile}\_iftrue \.printintro
141       \_else \.printwarn{File with introduction text \introfile\_space not found}\_fi
142 %
143       \_isfile{\fmtfile}\_iftrue \_input{\fmtfile}
144       \_else \.printwarn{File with format info \fmtfile\_space not found}\_fi
145       \_isfile{\notesfile}\_iftrue \_input{\notesfile}
146       \_else \.printwarn{File with notes \notesfile\_space not found}\_fi
```

```

147     \cs{bpr!#1}
148     \.bibleinput{\txsfile}
149     \.chapafter % material after the last chapter
150     \cs{bpo!#1}
151     \endgroup
152     \ea \.processbooksB
153     \_fi
154 }
155 \_nspublic \processbooks ;

```

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

op-bible.opm

```

161 \_def\.newbook#1{\_vfil\_supereject
162   \_let\.prelinkB=\.currbook \.chapnum=0
163   \_def\.prelinkC{0}\_def\.prelinkV{0}
164   \_global\_headline={\_hfil \_ea\.setheadline\_ea{\.btit}}
165   \_line{\_hss\.bookfont\.btit\_hss}
166   \_par\_nobreak\_medskip
167 }

```

`\.setheadline{<book-title>}` sets `_headline`. It is re-set for each new book by `\.newbook`.

The `\bibname` can be defined by user as a name of the translating variant of the Bible. If it is not defined then it is empty by default.

op-bible.opm

```

176 \_def\.setheadline#1{\_global\_headline={\_headfont
177   \_ifodd\_pageno
178     \_rlap{\_it\bibname\_hss}%
179     \_hfil \_the\_pageno\_hfil
180     \_hbox to\_lrmargin{\_hss\_bf#1\_ifx\_botmark\_else\_space \_botmark\_fi}%
181     \_kern-\_lrmargin
182   \_else
183     \_kern-\_lrmargin
184     \_hbox to\_lrmargin{\_bf#1 \_firstmark\_hss}%
185     \_hfil \_the\_pageno\_hfil
186     \_llap{\_hss\_it\bibname}%
187   \_fi
188 }
189 }
190 \_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.

op-bible.opm

```

203 \_def\.checknochapbooks {%
204   \_ifx\nochapbooks\_undefined
205     \.printwarn{\_noexpand\nochapbooks (boks without chapters) undefined.}%
206     \_def\nochapbooks{}%
207   \_else \_edef\nochapbooks{\_space\nochapbooks\_space}\_fi
208 }

```

3 Book titles

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

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

op-bible.opm

```

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

```

The `\BookException <a-mark> {<code>}` macro adds the `<code>` to the `\bex!<a-mark>` macro. It is used in `\processbooks` loop in the group before files are read. You can redefine some filenames or something more special here.

Macros `\BookPre` $\langle a\text{-mark} \rangle$ $\{\langle code \rangle\}$ and `\BookPost` $\langle a\text{-mark} \rangle$ $\{\langle code \rangle\}$ are defined similarly. They add $\langle code \rangle$ to the `\bpr!` $\langle a\text{-mark} \rangle$ and to the `\bpo!` $\langle a\text{-mark} \rangle$ macros respectively.

op-bible.opm

```
237 \_outer\_long\_def\BookException #1 #2{\.myaddto{bex!#1}{#2}}
238 \_outer\_long\_def\BookPre      #1 #2{\.myaddto{bpr!#1}{#2}}
239 \_outer\_long\_def\BookPost     #1 #2{\.myaddto{bpo!#1}{#2}}
240
241 \_nspublic \BookTitle \BookException \BookPre \BookPost ;
```

The `\ChapterPre` $\{\langle code \rangle\}$ and `\ChapterPost` $\{\langle code \rangle\}$ inserts $\langle code \rangle$ before each chapter and after each chapter. The $\langle data \rangle$ are the same for each chapter, it does not vary depending on the Book or Chapter number.

op-bible.opm

```
249 \_long\_def\ChapterPre #1{\_def\chapbefore{#1}}
250 \_long\_def\ChapterPost #1{\_def\chapafter{#1}}
251
252 %\_outer\_def\ChapterPre {\ChapterPre}
253 %\_outer\_def\ChapterPost {\ChapterPost} % be done at the end of this file
```

4 Actions

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

`\alist!` $\langle full\text{-}vref \rangle$ is the list of actions associated with the verse $\langle full\text{-}vref \rangle$. The $\langle full\text{-}vref \rangle$ is full reference to the verse in the format $\langle book\text{-}mark \rangle / \langle chapter\text{-}num \rangle : \langle verse\text{-}num \rangle$

`\.newaction` $\{\langle full\text{-}vref \rangle\} \{\langle action\text{-}body \rangle\}$ allocates new action.

op-bible.opm

```
273 \_def\newaction#1#2{%
274   \_unless\_ifcurname alist!#1\_endcurname \_sdef{alist!#1}{\_fi
275   \_ea\_addto\_csname alist!#1\_endcurname{#2}%
276 }
```

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

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

If $\langle text \rangle$ does not exist then $\langle fail \rangle$ is processed. It can report failed $\langle text \rangle$ by the `\.text` macro.

op-bible.opm

```
289 \_def\replpre#1#2#3{%
290   \_ifx\^#2\_def\tmp{#1}\_ea\_ea\_ea\_def\_ea\_ea\_ea\buff\_ea\_ea\_ea{\_ea\_.tmp\buff}%
291   \_else
292     \_def\replpredo##1#2##2\_end{%
293       \_ifx\_end##2\_end \_def\text{#2}#3% <fail>
294       \_else \replsave ##1#1{#2}##2\_end \_fi
295     }%
296     \_def\replsave##1#2\_end{\_def\buff{##1}}%
297     \_ea\replpredo\buff#2\_end
298   \_fi
299 }
```

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

op-bible.opm

```
309 \_def\replprepost#1#2#3#4{%
310   \_def\replprepostdo##1#1##2\_end{%
311     \_ifx\_end##2\_end \_def\text{#1}#4% <fail>
312     \_else \replsave ##1#2#1#3##2\_end \_fi
313   }%
314   \_def\replsave##1#1\_end{\_def\buff{##1}}%
315   \_ea\replprepostdo\buff#1\_end
316 }
```

5 The \Note macro

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

op-bible.opm

```
331 \_def\.gentovref#1{\.currbook/\.gentovrefA#1-\end}
332 \_def\.gentovrefA#1-#2\end{#1}
```

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

op-bible.opm

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

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

op-bible.opm

```
350 \_def\.transformword#1{%
351   \_ifcsname v!\tmark!#1\_endcsname \_lastnamedcs
352   \_else #1\_fi
353 }
```

`\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 saves it to `\.fullvref`.
- If the verse number of $\langle full-vref \rangle$ is zero, we want to insert the note-text before the chapter. This is one by the `\.NoteB` macro.
- Allocates new $\langle note-num \rangle$, i.e. `\.notenun` 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 text of a previous `\Note`.

op-bible.opm

```
399 \_newcount\.notenun
400 \_def\.Note #1 #2{%
401   \_edef\.fullvref{\.gentovref{#1}}%
402   \_ea\.isversezero\.fullvref\_iftrue
403     \_ea\.NoteB
404   \_else
405     \_incr\.notenun
406     \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
407     \_def\.tmp{#1}\_sedef{notepre!\_the\.notenun}{\_ea\.renumlabel\.fullvrefm\_relax}%
```

```

408 \_ifx\.\nextww\undefined
409 {\_def\.\printwarn##1{\_xdef\.\tword{\.\transformword{#2}}}%
410 \_else \_xdef\.\tword{\.\nextww}\_fi
411 \_afterfi{\_isnextchar={\.\NoteA}{\.\NoteA={}}}%
412 \_fi
413 }
414 \_def\.\NoteA=#1#2% #2 separated by \par or \_par:
415
416 {%
417 \_sdef{notetext!\_the\.\notenumber}{\_ignorespaces#2}%
418 \_sedef{noteref!\_the\.\notenumber}{\.\fullvrefm}%
419 \_ifx\.\nextww\undefined
420 \_ifx^#1^\_sdef{pword!\_the\.\notenumber\_ea}\_ea{\.\tword}\_else \_sdef{pword!\_the\.\notenumber}{#1}\_fi
421 \_else
422 \_sdef{pword!\_the\.\notenumber\_ea}\_ea{\.\nextwwA}%
423 \_let\.\nextww=\_undefined \_let\.\nextwwA=\_undefined
424 \_fi
425 \_reducetword
426 \_ea\.\addNote\_expanded{\.\fullvrefm}{\_the\.\notenumber}{\.\tword}}%
427 }
428 \_def\.\addNote#1#2#3{%
429 \_ifx^#3^\_tword is empty
430 \_edef\.\tmp{\_cs{notepre!#2}}%
431 \_ea \.\isdivisin\.\tmp-\_iftrue
432 \_newaction{#1}{\.\replpre{\.\doNote{#2}}{}}}%
433 \_else
434 \_newaction{#1}{\\_addto\.\prebuff{\.\doCNote{#2}}{}}}%
435 \_fi
436 \_else
437 \_newaction{#1}{\.\replpre{\.\doNote{#2}}{#3}{\.\notefail{#2}}}%
438 \_fi
439 }
440 %\_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.

op-bible.opm

```

449 \_def\.\NoteB #1% #1 separated by \par or \_par
450
451 {%
452 \_sdef{chapnote!\.\fullvref}{\_ignorespaces#1}%
453 }
454 \_def\.\isversezero#1/#2:#3\_iftrue{\_ifnum #3=0 }

```

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

op-bible.opm

```

467 \_def\.\renumlabel#1/#2\_relax#2%
468 \_ea\.\isdivisin\.\tmp-\_iftrue --\_ea\.\renumlabelA\.\tmp\_relax#2\_relax \_fi
469 }
470 \_def\.\renumlabelA#1:#2-#3\_relax#4:#5\_relax{%
471 \_iscolonin#3:\_iftrue #3\_else \_the\_numexpr#5+#3-#2\_relax \_fi
472 }

```

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

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

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

op-bible.opm

```

487 \_def\.\notefail#1{%
488 \.\printwarn{\_csstring\\Note: \.\currverse: The text "\_unexpanded\_ea{\.\text}" not found}%
489 \.\replpre{\.\doNote{#1}}{}}}% \Note is registered with the beginning of the verse
490 }

```

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

```

502 \def\prevnotepre{}
503 \def\doNote#1#2{%
504   \edef\tmpb{\_cs{notepre!#1}}}%
505   \notelog{\_space\_space\_csstring\\Note\tmpb\_space{#2}={\_cs{pword!#1}} (#1)}%
506   \noteinsert{%
507     {\_bf\_ifx\prevnotepre\tmpb\_else\tmpb\_enskip\_glet\prevnotepre=\tmpb\_fi
508     \trymakedest{n:\_cs{noteref!#1}}}%
509     \edef\tmpb{\_csname pword!#1\_endcsname}%
510     \_ifx\tmpb\_empty\_else
511       \addto\tmpb{.}\punctpword
512       \ea\upcasefirst\tmpb\_space
513     \_fi
514   }% end of \bf
515   \_cs{notetext!#1}}}%
516   {\notecolor#2}%
517 }
518 \def\_printfnotemark{}
519 \def\_textindent#1{\_noindent}

```

op-bible.opm

```
529 \_def\_.upcasefirst #1{\_uppercase{#1}}
```

op-bible.opm

```
537  \_def\.punctpword{\_replstring\.tmpb{!..}{!}\_replstring\.tmpb{?.}{?}}
```

op-bible.opm

```

548 \_def\doCNote #1{%
549   \edef\tmpb{\_csname pword!#1\_endcsname}%
550   \_notelog{\_space\_space \_csstring\Note \tmpb\_space }={\_cs{pword!#1}} (#1)%
551   \edef\prevnotepre{\_cs{notepre!#1}}%
552   \_ifx\tmpb\_empty \_else
553     \_addto\tmpb{.}\_punctpword
554     \edef\tmpb{\_noexpand\_bf \_ea\_upcasefirst\tmpb\_noexpand-}%
555     \_ea\_addto \_ea\_.Cnotetext \_ea{\tmpb}%
556   \_fi
557   \_ea\_ea\_ea\_addto\_ea\_ea\_ea\_.Cnotetext\_ea\_ea\_ea{\_csname notetext!#1\_endcsname}%
558 }
559 \_def\printCnote{%
560   \_ifx\_.Cnotetext\_empty \_else
561     \_noteinsert{%
562       {\_bf \_ea\_.nobook\_.currverse\_relax \_.trymakedest{n:\_.currverse}} \_.Cnotetext
563     }%
564   \_fi
565 }
566 \_def\nobook #1/#2\_relax {#2} % only chapter:verse is printed

```

op-bible.opm

```

575 \_def\.reducetword{}
576 \_def\.mergednotes{\_def\.reducetword{\_def\.tword{}}}
577 \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`.

op-bible.opm

```
590 \_let\.notelog=\_wlog
```

6 Inserting data from format files

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

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

`\fmtins {<gen-vref>}{<text>}{<what>}` inserts `<what>` after `<text>` in the verse. If `<text>` is not found then `<what>` is inserted like `\fmtpre` does it

All these commands allocate new action using `\.newaction`.

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

op-bible.opm

```
607 \_def\.fmtpre#1#2{\.newaction{\.gentovref{#1}}{\_addto\.fmtprebuff{#2}}}
608 \_def\.fmtpreind#1#2{\.newaction{\.gentovref{#1}}{\_addpre\.preindbuff{#2}}}
609 \_def\.fmtadd#1#2{\.newaction{\.gentovref{#1}}{\_addto\.buff{#2}}}
610 \_def\.fmtins#1#2#3{\.newaction{\.gentovref{#1}}{\.replprepost{#2}{#3}{\.fmtfail{#3}}}}
611 \_def\.fmtfail#1{\.fmtwarn\_addto\.fmtprebuff{#1}}
612 \_def\.fmtwarn{\.printwarn{\_string\fmtins: \.currverse: The text "\.text" not found}}
613 \_def\.addpre#1#2{\_ea\.addpreA \_ea{#1}{#2}#1}
614 \_def\.addpreA #1#2#3{\_def#3{#2#1}}
615
616 \_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.

op-bible.opm

```
625 \_newdimen\centermargin \centermargin=4em
626 \_def\.begcenter{\_par \_ifnum\_lastpenalty<10000 \_medskip \_fi
627   \_bgroup
628   \_def\.centeringmode{y}
629   \_parindent=0pt
630   \_leftskip=\centermargin plusifill
631   \_rightskip=\leftskip
632 }
633 \_def\.endcenter{\_par
634   \_ifx\.centeringmode\_undefined
635     \.printwarn{\_noexpand\endcenter ignored: no \_noexpand\begcenter precedes}
636   \_else \_egroup \_medskip \_fi
637 }
638 \_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`.

op-bible.opm

```
649 \_newifi\_ifopb_firstverse
650
651 \_def\.ind#1{\_unless \_ifopb_firstverse \_par \_else \_hskip-\_parindent \_fi
652   \_noindent
653   \_hskip#1\_iindent \_spacefactor=1001 }
```

`\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 `\fmtprepost` or `\fmtins` and using `\ind{_the_tmpnum}`.


```

663 \_def\fmtpoetry#1#2{\_def\tpa{#1}\fmtpoetA #2\_end}
664 \_def\fmtpoetA #1/{\_def\tpb{#1}\_tmpnum=1 \fmtpoetB}
665 \_def\fmtpoetB #1{\_ifx/#1 \_incr\_tmpnum \_ea\fmtpoetB \_else \_afterfi{\fmtpoetC#1}\_fi}
666 \_def\fmtpoetC #1{%
667   \_expanded{\_ifx\tpb\_empty \_noexpand\fmtpreind{\tpa}\_else
668     \_noexpand\fmtins{\tpa}{\tpb}\_fi{\_noexpand\_ind{\_the\_tmpnum}}}%
669   \_ifx\_end#1 \_else \_afterfi{\fmtpoetA#1}\_fi
670 }
671 \_nspublic \ind \fmtpoetry ;
672
673 \_def\fmtfont#1#2#3{%
674   \newaction{\gentovref{#1}}{\replprepost{#2}{\bgroup#3}{\egroup}{\fmtwarnf}}
675 \_def\fmtwarnf{\printwarn{\string\fmtfont: \currverse: The text "\text" not found}}
676 \_nspublic \fmtfont ;

```

7 Printing verses from .txs files

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

The `\processline <chapter>:<verse><space><verse-text>^^J` is repeatedly processed.

```

688 \_eoldef\processline#1{\processverse \currbook/#1\_end}

```

`\processverse <full-vref><space><verse-text>_end` does

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

```

703 \_newcount\chapnum
704 \_def\processverse #1 #2\_end{%
705   \_xdef\currverse{#1}%
706   \preparechapverse #1
707   \let\prelinkV=\currversenum
708   \_gdef\buff{#2}\_gdef\fmtprebuff{}\_gdef\preindbuff{}\_gdef\prebuff{}\_gdef\Cnotetext{}%
709   \_ifx\verseto\_empty \_csname alist!#1\_endcsname \_else
710     \_for num \versefrom..\verseto \_do{\_csname alist!\currbook/\currchapnum:#1\_endcsname}%
711   \_fi
712   \_ifnum\currchapnum=\chapnum \_else
713     \_ifnum\chapnum>1 \chapafter \_fi
714     \let\prelinkC=\currchapnum \chapnum=\currchapnum\_relax
715     \chapbefore \_fi
716   \printverse
717 }
718 \_def\preparechapverse #1/#2:#3 {\_def\currchapnum{#2}%
719   \_def\verseto{}}%
720 \_isdivisin #3-\_iftrue \defversefromto #3\_end
721 \_else \_def\currversenum{#3}\_glet\currversetext=\currversenum
722 \_fi
723 }
724 \_def\defversefromto #1-#2\_end{%
725   \_def\versefrom{#1}\_def\verseto{#2}%
726   \_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.

```

734 \_def\prepareversetext{}
735 \_def\cnvtext#1#2{\_addto\prepareversetext{\replstring\buff{#1}{#2}}}
736 \_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.

op-bible.opm

```

752 \_def\.printverse{%
753   \_fmtprebuff % material accumulated by \fmtpre
754   \_ifnum\.currversenum=1 \_firstversetrue \.printbeforefirst \_fi
755   \_quitvmode \_mark{\.currchapnum:\.currversetext}%
756   \_ifx\.verseto\_empty \_trymakedest{v:\.currverse}%
757   \_else \_forunum \.versefrom..\verseto \_do{%
758     \_wlog{xxxxx v:\.currbook/\.currchapnum:##1}\_trymakedest{v:\.currbook/\.currchapnum:##1}}%
759   \_fi
760   \_preindbuff
761   \_raise5pt\.hboxorllap{\_unless\_ifnum\.currversenum=1 \_markfont\.currversetext\,\_fi}%
762   \_firstversefalse
763   \_prepareversetext
764   \_prebuff\.printCnote\_buff \_space
765 }
766 \_def\.hboxorllap{\_ifnum\_spacefactor=1001 \_ea\_llap \_else \_ea\_hbox \_fi}
767
768 \_def\.printbeforefirst{%
769   \_par\_nobreak \_medskip
770   \_trychapnote
771   \_setbox0=\_vtop{\_kern-1.5ex \_ewref\_sxdef{{ch!\.currbook/\_the\.chapnum}{\_string\_mypage}}
772     \_hbox{\_setfontsize{at50pt}\_bf\LiRed\_the\.chapnum}}
773   \_dp0=0pt
774   \_tmpdim=\_lrmargin
775   \_advance\_tmpdim by4pt
776   \_ifnum\_the\.chapnum>9 \_advance\_tmpdim by19pt \_fi
777   \_ifodd\_trycs{ch!\.currbook/\_the\.chapnum}{0}
778     \_moveright\_tmpdim \_line{\_hss\_box0}
779   \_else \_moveleft\_tmpdim \_box0 \_fi
780   \_nobreak \_vskip-\_medskipamount
781   \_nobreak \_nointerlineskip \_noindent
782 }

```

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

op-bible.opm

```

789 \_def\.trychapnote{%
790   \_ifcsname chapnote!\.currbook/\_the\.chapnum:0\_endcsname
791     \.printchapnote{\_cs{chapnote!\.currbook/\_the\.chapnum:0}}\_fi
792 }
793 \_def\.printchapnote #1{\_par
794   {\_leftskip=\_parindent plus1fill \_rightskip=\_leftskip \_noindent\_it #1\_par}
795   \_medskip
796 }
797 \_nspublic \printchapnote ;

```

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

op-bible.opm

```

805 \_def\.chapbefore{\_bigskip} \_def\.chapafter{}

```

8 Bible references

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

- `\.ltextP` ... the text before a link specification (given in "...")
- `\.ltextB` ... the book mark followed by ~
- `\.ltextC` ... the chapter number followed by :
- `\.ltextV` ... the verse number

- `\.ltextS` ... sub-verse identifier (a if there is a verse 4a)
- `\.ltextF` ... the -- if the $\langle from \rangle$ - $\langle to \rangle$ format is given
- `\.ltextN` ... the $\langle to \rangle$ part from the $\langle from \rangle$ - $\langle to \rangle$ format.

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

op-bible.opm

```

833 \_def\.linktext{\.ltextP\.ltextB\.ltextC\.ltextV\.ltextS\.ltextF\.ltextN}
834 \_def\.bref #1>{\_let\.brefH=\_relax \_def\.linkspec{#1}\_isnextchar{"\brefA}{\brefA"}#1>}
835 \_def\.brefA"#1"{\_def\.ltextP{#1}%
836 \_isnextchar{ }{\_addto\.ltextP{~}\_afterassignment\.brefB\_let\.next= }%
837 {\_isnextchar_{}\_def\.brefH{\_afterassignment\.brefB\_let\.next= }\brefB}}%
838 }
839 \_def\.brefB #1>{% #1 is link-spec
840 \_def\.ltextB{\_def\.ltextC{\_def\.ltextF{\_def\.ltextN{\_
841 \_isspacein #1 \_iftrue
842 \_iscolonin #1:\_iftrue \.brefBookChapterVerse #1>%
843 \_else \.brefBookChapter #1>\_fi
844 \_else \_iscolonin #1:\_iftrue \.brefChapterVerse #1>%
845 \_else \.brefVerse #1>%
846 \_fi\_fi
847 \_def\.linkpre{v}%
848 \_isnextchar n{\_def\.linkpre{n}\.brefC}%
849 {\_isnextchar g{\_def\.linkpre{g}\.brefC}%
850 {\_isnextchar a{\_def\.linkpre{a}\.brefC}%
851 {\_isnextchar i{\_def\.linkpre{i}\.brefC}{\brefD}}}%
852 }
853 \_def\.brefC{\_afterassignment\.brefD \_let\.next= }
854
855 \_def\.brefBookChapterVerse #1 #2:#3>{\_def\.ltextB{#1}\.brefChapterVerse #2:#3>}
856 \_def\.brefBookChapter #1 #2>{\_def\.ltextB{#1~}%
857 \_isinlist\nochapbooks{ #1 } \_iftrue
858 \_def\.ltextC{\_let\.ltextCin=\.ltextnCin \_afterfi{\.brefVerse #2>}}%
859 \_else \_afterfi{\.brefChapter #2>}\_fi}
860 \_def\.brefChapterVerse #1:#2>{\_def\.ltextC{#1:}\.brefVerse #2>}
861 \_def\.brefVerse #1>{%
862 \_isdivisin #1-\_iftrue \.brefFromTo #1>%
863 \_else \.versedef#1\_relax\_fi
864 }
865 \_def\.brefChapter #1>{%
866 \_isdivisin #1-\_iftrue \.brefFromTo #1>\_let\.ltextC=\.ltextV
867 \_else \_def\.ltextC{#1}\_fi
868 \_def\.ltextV{\_def\.ltextS{\_
869 }
870 \_def\.brefFromTo #1-#2>{\.versedef#1\_relax\_def\.ltextF{--}\_def\.ltextN{#2}}

```

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

op-bible.opm

```

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

```

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

op-bible.opm

```

886 \_def\.brefD{%
887 \_ifnum 0\.ltextV=0 \_def\.ltextV{\_fi
888 \_if a\.linkpre \_ifx\.ltextV\_empty \_else \_edef\.ltextC{\.ltextV:}\_def\.ltextV{\_fi\_fi
889 \_edef\.linkfspec{\_ea\.ltextBin\.ltextB~/\_ea\.ltextCin\.ltextC:/\_ea\.ltextVin\.ltextV:/}%
890 \.brefL
891 }
892 \_def\.ltextBin #1-#2/{\_ifx^#1^\.prelinkB \_else #1\_immediateassignment\_def\.prelinkB{#1}\_fi/}
893 \_def\.ltextCin #1:#2/{\_ifx^#1^\.prelinkC \_else #1\_immediateassignment\_def\.prelinkC{#1}\_fi:}
894 \_def\.ltextVin #1:#2/{\_ifx^#1^\.prelinkV \_else #1\_immediateassignment\_def\.prelinkV{#1}\_fi}
895 \_def\.ltextnCin #1:#2/{\_prelinkC: \_immediateassignment\_let\.ltextCin=\.ltextS}
896 \_let\.ltextS=\.ltextCin

```

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

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

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

```
914 \_newtoks\everybref
915 \_def\.oncebref{}
916 \_nspublic \everybref ;
```

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

`\.renumlinktext` $\langle full-vref-ori \rangle_relax\langle full-vref-modified \rangle_relax$ does re-calculation of the parts of the `\.linktext` macro.

The `\.linkfspecone` solves situation when chapter is given but no verse number: we must set the verse number to 1.

If the link destination is article, then the $\langle full-vref \rangle$ has reduced format $\langle book \rangle/\langle chapter \rangle$. If the link destination is introduction then the $\langle full-vref \rangle$ has more reduced format: $\langle book \rangle/$.

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 `\.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.

```
940 \_def\.brefL{%
941   \_edef\.linkfspecm{\_ea\.renumvref\.linkfspec\_relax}%
942   \_ifx\.linkfspec\.linkfspecm \_else
943     \_ea\_ea\_ea\.renumlinktext \_ea\.linkfspec \_ea\_relax \.linkfspecm \_relax
944     \_let\.linkfspec=\.linkfspecm
945   \_fi
946   \_ifx\.ltextV\_empty \_ifx\.ltextC\_empty \_else \_ea\.linkfspecone \.linkfspec\_end \_fi\_fi
947   \_if a\.linkpre\_relax \_ea\.linkfspecarticle \.linkfspec\_end \_fi
948   \_if i\.linkpre\_relax \_ea\.linkfspecintro \.linkfspec\_end \_fi
949   \_ifx \.ltextB\_empty \_else \_ea \.newltextB \.ltextB \_fi
950   \.linklog{\_sspace <\_unexpanded\_ea\_.linkspec}>\.linkpost = [\_.linkpre:\.linkfspec]%
951   {\_ifx\.brefH\_empty \.ltextP \_else \.linktext\_fi}}%
952   \.ensuredest \.createlink
953 }
954 \_def\.linkfspecone #1:#2\_end {\_def\.linkfspec{#1:1}\_def\.prelinkV{1}}
955 \_def\.linkfspecarticle #1/#2:#3\_end {\_def\.linkfspec{#1/#2}}
956 \_def\.linkfspecintro #1/#2\_end {\_def\.linkfspec{#1/}}
957
958 \_def\.renumlinktext #1/#2:#3\_relax #4/#5:#6\_relax{%
959   \_ifx\.ltextC\_empty \_else \_def\.ltextC{#5}\_fi
960   \_def\.ltextV{#6}%
961   \_ifx \.ltextN\_empty \_else
962     \_ifx \.ltextF \.ltextDD
963       \_isinlist \.ltextN{:}\_iftrue
964         \_ifcsname rn!\tmark!#1/\.ltextN\_endcsname \_edef\.ltextN{\_cs{rn!\tmark!#1/\.ltextN}}%
965         \_fi
966       \_else \_edef\.ltextN{\_the\_numexpr#6+\.ltextN-#3\_relax}\_fi
967     \_else \_let\.tmp=\_ignreit % \.ltextN is a list of verses, for example 7,9,13
968     \_ea\_foreach \.ltextN,\_do ##1,{\_edef\.tmp{\_tmp,\_the\_numexpr#6+##1-#3}}%
969     \_let \.ltextN=\.tmp
970   \_fi
971   \_fi
972 }
973 \_def\.ltextDD{--}
974
975 \_def\.newltextB #1~{\_edef\.ltextB{\_trycs{v!\tmark!#1}{#1}~}}
976
977 \_def \_sspace{\_space\_space\_space\_space}
978 \_def \.linkpost{\_if v \.linkpre \_else \.linkpre\_fi \_space}
```

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

op-bible.opm

```
984 \def\tracinglinks{\let\linklog=\wlog}
985 \def\notracinglinks{\let\linklog=\ignoreit}
986 \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.

op-bible.opm

```
997 \def\.createlink{%
998   \ifx\brefH\_empty \let\linktext=\ltextP\_fi
999   \ea\isprintedbook\linkfspec \iftrue
1000   \link[\linkpre:\linkfspec]{\ilinkcolor}{\linktext}%
1001   \else {\ilinkcolor\linktext}\_fi}%
1002 }
1003 \def\isprintedbook #1/#2\_iftrue{\_ifcsname pbook!#1\_endcsname}
1004 \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 `\.ensuredet` 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.

op-bible.opm

```
1024 \_newwrite\ .xrf
1025 \_immediate\_openout\ .xrf=\_jobname.xrf
1026 \_openref
1027
1028 \def\.ensuredet{\_immediate\_write\ .xrf{\_string\_sdef{\linkpre:\linkfspec}{}}}
1029 \_refdecl{
1030   \_isfile{\_jobname.xrf}\_iftrue \_input{\_jobname.xrf}\_fi^^J
1031   \def\.Xdest#1{\_ifcsname pg:#1\_endcsname \_sxdef{pg:#1}{\_ea\_usessecond\_currpage}\_fi^^J
1032   \def\ .mypage{\_ea\_usessecond\_currpage}
1033 }
1034 \def\ .trymakedest#1{%
1035   \_ifcsname #1\_endcsname \_dest[#1]\_ea\_glet\_csname #1\_endcsname \_undefined \_fi
1036   \_ewref\ .Xdest{#1}%
1037 }
```

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

op-bible.opm

```
1045 \def\pg{%
1046   \_ifcsname pg:\linkpre:\linkfspec\_endcsname
1047   {\_edef\linktext{\_cs{pg:\linkpre:\linkfspec}}\_let\brefH=\_relax \.createlink}%
1048   \else {\Red ??}\_fi
1049   \_immediate\_write\ .xrf{\_string\_sdef{pg:\linkpre:\linkfspec}{??}}%
1050 }
1051 \_nspublic \pg ;
```

9 Language variants

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

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

op-bible.opm

```
1063 \_newcount\.numvariants
1064 \_def\.variants{\_tmpnum=0 \_afterassignment\.variantsA \.numvariants}
1065 \_def\.variantsA{%
1066   \_ifnum\_tmpnum<\.numvariants
```

```

1067     \advance\_tmpnum by1
1068     \afterfi{\.variantsB{\_the\_tmpnum}}%
1069     \_fi
1070 }
1071 \_def\.variantsB#1#2{%
1072     \_ifnum#1=1 \_gdef\tmarkA{#2}\_sxddef{var!1}{#2}%
1073     \_else \_sxddef{var!#1}{#2}%
1074     \_fi
1075     \.variantsA
1076 }
1077 \_nspublic \variants ;

```

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

```

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

```

op-bible.opm

```

1094 \_def\.vdef#1{\_def\.tmp{#1}%
1095     \_ifcsname v!\_trycs{var!2}{!}\.tmp\_endcsname
1096     \.printwarn{\_noexpand\vdef used secondly for phrase {\.tmp}, ignored}\_fi
1097     \_tmpnum=1 \_ea\.vdefA
1098 }
1099 \_def\.vdefA{%
1100     \_ifnum\_tmpnum<\.numvariants
1101         \advance\_tmpnum by1
1102         \afterfi{\.vdefB{\_the\_tmpnum}}%
1103     \_fi
1104 }
1105 \_def\.vdefB#1#2{\_def\.tmpa{}}%
1106     \_ifx\.vdef#2\_def\.tmpa{#2}\_fi
1107     \_ifx\.tmpa\_empty
1108         \_ifx^#2\_else
1109             \_unless \_ifcsname v!\_cs{var!#1}!\.tmp\_endcsname
1110                 \.sedef{v!\_cs{var!#1}!\.tmp}{\_ifx"#2\.prevcs{#1}\.tmp \_else#2\_fi}%
1111             \_fi\_fi
1112             \_ea\.vdefA
1113         \_else \_errmessage{\_string\vdef: too few parameters. To be read again: \_string#2}%
1114         \_ea\.tmpa
1115     \_fi
1116 }
1117 \_def\.prevcs #1#2{\_ifnum#1=2 #2\_else \_cs{v!\_cs{var!\_the\_numexpr#1-1\_relax}!#2}\_fi}
1118
1119 \_nspublic \vdef ;

```

\x/**<phrase>**/ expands to **\v!**<mark>!**<phrase>** if such control sequence is defined else it expands simply to **<phrase>** using **\xA**. The <mark> is actual value of the **\tmark** macro.

Note that if **\tmark** expands to <t-markA> (used in the **\variants** macro), then the **\v!**<mark>!**<phrase>** is not defined and the **\x** macro expands to the **<phrase>** directly.

\xA **<phrase>**/ expands to **<phrase>** and prints warning, if **\tmark** is not the first <t-markA>.

op-bible.opm

```

1132 \_def\.x/#1/{\_trycs{v!\_tmark!#1}{\_xA#1/}}
1133 \_def\.xA#1/{#1\_ifx\tmarkA\_undefined \_else \_ifx\tmark\tmarkA \_else
1134     \.printwarn{\_string\x/#1/ -- this phrase is undefined by \_csstring\vdef}%
1135     \_fi\_fi
1136 }
1137 \_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.


```

1150 \_def\.\ww{%
1151   \_ifx\.\varnum\_undefined \.setvarnum \_fi
1152   \_tmpnum=0
1153   \_ifx\.\nextww\_undefined \_ea\.\wwA
1154   \_else \.printwarn{Only single \_csstring\.\ww must be before \_csstring\.\Note}%
1155   \_ea\.\wwB \_fi
1156 }
1157 \_def\.\wwA#1#2 {\_advance\_tmpnum by1
1158   \_def\.\nextww{#1}\_def\.\nextwwA{#2}%
1159   \_ifx\.\nextwwA\_empty \_let\.\nextwwA=\.\nextww \_else \_ea \.redefwwA #2\_end \_fi
1160   \_ifnum\.\varnum=\_tmpnum \_ifnum\_tmpnum<\.numvariants \_ea\_ea\_ea \.wwB \_fi
1161   \_else \_ea \.wwA \_fi
1162 }
1163 \_def\.\wwB#1 {\_advance\_tmpnum by1
1164   \_ifnum\_tmpnum<\.numvariants \_ea\.\wwB \_fi
1165 }
1166 \_def\.\redefwwA =#1\_end{\_def\.\nextwwA{#1}}
1167
1168 % \_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.

```

1182 \_newtoks\.\switchD
1183 \_def\.\switch {\_let\.\switchN=\.switchA \_suppressoutererror=1 \.switchN}
1184 \_long\_def\.\switchA #1#2{\.switchD={#2\_let\.\switchN=\.switchI}%
1185   \_ifx\_relax#1\_relax \_the\.\switchD
1186   \_else \_foreach #1,\_do ##1,{\_def\tmp{##1}\.switchC}%
1187   \_fi
1188   \_futurelet\.\next\.\switchB
1189 }
1190 \_def\.\switchB{\_ifx\.\next\_bgroup \_ea\.\switchN \_else \_suppressoutererror=0 \_fi}
1191 \_long\_def\.\switchI #1#2{\_futurelet\.\next\.\switchB}
1192 \_def\.\switchC{\_ifx\tmp\tmark \_the\.\switchD \_fi}
1193
1194 \_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.

```

1202 \_def\.\setvarnum{\_gdef\.\varnum{0}%
1203   \_ifnum\.\numvariants=0 \_gdef\.\varnum{1}\_wlog{There is only single language variant (1)}%
1204   \_else
1205     \_tmpnum=0
1206     \_loop
1207       \_advance\_tmpnum by1
1208       \_ea\_ifx \_csname var!\_the\_tmpnum\_endcsname \tmark \_xdef\.\varnum{\_the\_tmpnum}\_fi
1209       \_ifnum\_tmpnum<\.numvariants \_repeat
1210         \_ifnum \.varnum=0 \_errmessage{\_noexpand\tmark isn't set, \_noexpand\.setvarnum failed}%
1211         \_else \_wlog{Language variant set by \_string\tmark{\tmark} (\.varnum)}\_fi
1212     \_fi
1213 }

```

`\renum <book-mark> <chapter-num>:\<verse-num> = <t-mark> <chap-num>:\<from>-\<to>` does

```

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

```

```

1227 \_def\.\renum #1 #2:#3 = #4 #5:#6-#7 {%
1228   \_tmpnum=#3\_relax
1229   \_for num #6..#7 \_do {\_sxdef\rn!#4!#1/#2:\_the\_tmpnum}{#5:#1}\_incr\_tmpnum}%

```



```

1230 }
1231 \nspublic \renum ;

```

10 Inserting notes to the page

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

op-bible.opm

```

1240 \newinsert \.noteins
1241 \skip\_.noteins=\bigskipamount % noterule height
1242 \count\_.noteins=500 % two columns
1243 \dimen\_.noteins=\maxdimen % full page of notes allowed

```

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

op-bible.opm

```

1256 \def\.noteinsert #1{\insert\.noteins{%
1257   \.noteset
1258   \vbox to\ht\strutbox{}\nobreak \vskip-\baselineskip
1259   #1\unskip\par \nobreak \vskip-\baselineskip
1260   \hbox{\_lower\_dp\strutbox\vbox{}}
1261   \penalty0
1262 }}
1263 \def\.noteset{\Heros\cond \scalemain \typoscale[800/800] % Heros condensed 80%
1264   \Black \nobreak
1265   \widowpenalty=20 \clubpenalty=20
1266   \leftskip=0pt \rightskip=0pt \parfillskip=0pt plus1fill
1267   \parindent=0pt
1268   \lineskiplimit=-3pt
1269   \hsize=.5\hsize \advance\hsize by-1em\relax % two columns
1270   \everypar{}
1271 }

```

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

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

op-bible.opm

```

1292 \_addto\_pagecontents{%
1293   \_ifvoid\_.noteins \_else
1294     \vskip\_skip\_.noteins \noterule
1295     \setbox\_.noteins=\vbox{\_penalty0 \unvbox\_.noteins \vfil}
1296     \splittopskip=12pt
1297     \setbox0=\vsplit\_.noteins to0pt % adding \splittopskip to \.noteins
1298     \def\_Ncols{2}
1299     \dimen0=.5\ht\_.noteins \setbox6=\_box\_.noteins
1300     \vskip\_splittopskip
1301     \_balancecolumns
1302     \_fi
1303     \_unless\_ifvoid\_.botins \unvbox\_.botins
1304     \_else \vskip 0pt plus1filll minus8pt \_fi
1305   }
1306 \_def \noterule {\_kern-3pt {\Black \hrule width\_hsize}\_kern 2.6pt }

```

11 Inserting images and articles to the page

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

op-bible.opm

```
1318 \_newinsert\.botins
1319 \_def\.botinsert{\_setbox0=\_vbox\_bgroup}
1320 \_def\.endbot{\_par\_egroup
1321   \_insert\.botins{\_splittopskip=0pt \_penalty100
1322     \_hrule height0pt \_nobreak\_medskip\_bigskip \_unvbox0
1323   }%
1324 }
1325 \_skip\.botins=\_zoskip    % no space added when a topinsert is present
1326 \_count\.botins=1000      % magnification factor (1 to 1)
1327 \_dimen\.botins=\_maxdimen % no limit per page
```

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

op-bible.opm

```
1340 \_def\.putImage #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1341   \_edef\.fullvref{\.gentovref{#1}}%
1342   \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
1343   \_ea\.newaction\_ea{\.fullvrefm}{\.doImage{#2}[#4] (#6){#7}}%
1344 }
1345 \_def\.doImage #1[#2] (#3)#4{% {Title}[label] (params){image-file.pdf}
1346   \.botinsert
1347   \.botTitle{#1}[#2]%
1348   \_kern3pt \_nobreak
1349   \_hbox{\picw=\hsize #3\inspic{#4}}%
1350   \.endbot
1351 }
1352 \_def\.botTitle#1[#2]{\_hbox{\_captionfont
1353   \_ifx^#2\_else \.botDest{#1}[#2]\_fi
1354   \_rlap{\Grey \_vrule height1.2em depth.5em width\_hsize}\White\_kern12pt #1}%
1355 }
1356 \_picdir={images/}
1357 \_def\.botDest#1[#2]{\_label{#2}\_wlabel{#1}}
1358
1359 \_nspublic \putImage ;
```

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

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

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

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

op-bible.opm

```
1383 \_newcount\.articlenum
1384 \_def\.putArticle #1 #2#3[#4]#5(#6){% chap:verse {Title} [number] (params)
1385   \_edef\.fullvref{\.gentovref{#1}}%
1386   \_edef\.fullvrefm{\_ea\.renumvref\.fullvref\_relax}%
1387   \_ea\.newaction\_ea{\.fullvrefm}{\.doArticle{#2}[#4] (#6)}%
1388 }
1389 \_nspublic \putArticle ;
```

The `\.doArticle {<Title>} [<label>] (<params>)` 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 `\forloop` loop. These pairs are completed to blocks with LightGrey background. These blocks divided by `\break` are inserted into `_botinsert`.

op-bible.opm

```

1406 \_def\_.doArticle#1[#2](#3){% {Title}[number](params)
1407   \_incr\_.articlenum
1408   \_botinsert
1409   \_def\_.botDest##1[##2]{\_.trymakedest{a:\_.currbook/##2}}
1410   \_parindent=12pt \_iindent=\_parindent
1411   \_setbox0=\_vbox{\_hsize=.458\_hsize \_emergencystretch=1em
1412     \_hbadness=6000 \_baselineskip=\_dimexpr\_baselineskip plus1pt
1413     \_def\Article[#1]{\_endinput}
1414     \_penalty0
1415     \_long\_def\_.searcharticle##1\Article[#2]{
1416       \ea\_.searcharticle \_input \articlefile \_relax}
1417   \_splittopskip=12pt
1418   \_setbox1=\_vsplit0 to0pt % adding \splittopskip
1419   \_tmpdim=\_vsize \_advance\_tmpdim by-24pt % \_.botTitle height plus above/below skips
1420   \_ifdim 2\_tmpdim > \_ht0 \_tmpnum=1
1421   \_else
1422     \_tmpnum=\_.roundexpr{\_bp{\_ht0}/\_bp{1.333\_vsize}+0.999} % number of 2/3 pages
1423   \_fi
1424   \_multiply\_tmpnum by2 % number of columns
1425   \_edef\_Ncols{\_the\_tmpnum}
1426   \_dimen0=\_expr{1/\_Ncols}\_ht0 \_setbox6=\_box0 % height of each two-columns part
1427   \_setbox0=\_vbox{\_balancecolumns}
1428   \_tmpdim=\_ht0 \_advance\_tmpdim by1.2\_baselineskip
1429   \_setbox0=\_vbox{\_unvbox0 \_global\_setbox2=\_lastbox}
1430   \_setbox0=\_hbox{\_unhbox2
1431     \_forloop 1..\_Ncols \_do {\_unskip \_global\_setbox1##1=\_lastbox}}
1432     \_forloopstep -2: \_Ncols..1 \_do {
1433       \_hrule height0pt\_kern5pt\_nobreak\_vfill
1434       \_ifnum\_Ncols=##1 \_.botTitle{#1}[#2]\_else \_.botTitle{}[]\_fi
1435       \_kern3pt \_nobreak
1436       \_hbox to\_hsize{%
1437         \_rlap{\LightGrey \_vrule height\_tmpdim depth6pt width\_hsize}%
1438         \_kern\_parindent
1439         \_box1##1\_hss\_box1\_the\_numexpr##1-1
1440         \_kern\_parindent
1441       }
1442     }
1443   \_break
1444 }
1445 }
1446 \_def\_.roundexpr#1{\_ea\_ea\_ea\_.roundexprA\_expr{#1}\_relax}
1447 \_def\_.roundexprA#1.#2\_relax{\_ifnum#1=0 0\_else #1\_fi}

```

12 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 `\insertImage` saves the image in the box `\.spanpicbox`. The `_picwidth` of the image is calculated as $2 * (_hsize * \langle inner_margin \rangle)$. 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 `\endouput` 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.

op-bible.opm

```

1493 \_newbox \.spanpicbox
1494
1495 \_def\insertSpanImage #1#2[#3]#4(#5)#6{%
1496   \_par \_penalty0
1497   \_tmpdim=\_pagewidth
1498   \_advance\_tmpdim by-\_hoffset
1499   \_global\_setbox\.spanpicbox=\_hbox{\_picwidth=2\_tmpdim \_inspic{#6}}
1500   \_gdef\startinsertSpanImage {\insertBot {#1}[#3] (#5){\_copy\.spanpicbox \_kern-1.2ex}}
1501   \_doinsertSpanImage
1502 }
1503 \_def\doinsertSpanImage{%
1504   \_ifodd\_pageno
1505     \_glet\inspicbox=\inspicboxafter
1506   \_else
1507     \_ifdim \_dimexpr \_pagegoal-\_pagetotal > \_dimexpr \_ht\.spanpicbox+2em \_relax
1508     \_startinsertSpanImage
1509   \_else
1510     \_glet\inspicbox=\inspicboxafter
1511   \_fi
1512 }
1513 }
1514 \_let\inspicbox=\_useit
1515 \_def\inspicboxafter #1{%
1516   \_ifodd\_pageno
1517     \_startinsertSpanImage
1518   \_glet\inspicbox=\_useit
1519   \_fi
1520 }
1521 \_def \_endoutput{%
1522   \_ifvoid\.spanpicbox\_else \_addpicbox\_fi
1523   \_advancepageno
1524   {\_globaldefs=1 \_the\_nextpages \_nextpages={}}%
1525   \_ifnum\_outputpenalty>-20000 \_else\_dosupereject\_fi
1526 }
1527 \_def\addpicbox{\inspicbox{\insertBot{[]}() {\moveleft\_pagewidth\_box\.spanpicbox\_kern-1.2ex}}}

```

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

op-bible.opm

```

1537 \_long\_def\insertSpanText #1#2[#3]#4(#5)#6{%
1538   \_par \_penalty0
1539   \_tmpdim=\_pagewidth
1540   \_advance\_tmpdim by-\_hoffset
1541   \_setbox0=\_hbox to2\_tmpdim{\_hss\_vbox{\_hsize=2\_tmpdim
1542     \_leftskip=0pt \_rightskip=0pt \_relax \_kern3pt #6}\_hss}
1543   \_global\_setbox\.spanpicbox=
1544     \_hbox{\_rlap{\_White \_vrule width\_wd0 height\_ht0 depth\_dp0}\_box0}
1545   \_global\_ht\.spanpicbox=\_dimexpr\_ht\.spanpicbox-3pt\_relax
1546   \_gdef\startinsertSpanImage {\insertBot {#1}[#3] (#5){\_copy\.spanpicbox \_kern-1.2ex}}
1547   \_doinsertSpanImage
1548 }
1549 \_nspublic \insertSpanImage \insertSpanText ;

```

13 Inserting citations to the page

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

```
1561 \_def\putCite #1 #2{% chap:verse {text}
1562 \_edef\fullvref{\gentovref{#1}}%
1563 \_edef\fullvrefm{\_ea\renumvref\fullvref\_relax}%
1564 \_ea\newaction\_ea{\fullvrefm}{\dotopCite{#2}}%
1565 }
1566 \_nspublic \putCite ;
```

op-bible.opm

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

```
1578 \_newcount\citenum
1579 \_def\dotopCite #1{%
1580 \_topinsertnopar
1581 \_typosize[12/16]\_bi
1582 \_incr\citenum
1583 \_ifodd \_trycs{ct!\_the\citenum}{0}\_relax
1584 \_leftskip=.3\_hsize plus1fil \_parfillskip=0pt
1585 \_noindent
1586 \_rlap{\_hskip\_hsize \_kern-\_leftskip \_copy\lqqbox}\_hfill
1587 \_else
1588 \_let\quotedby=\_quotedbyright
1589 \_rightskip=.3\_hsize plus 1fil
1590 \_noindent \_llap{\_copy\lqqbox}%
1591 \_fi
1592 {\_printCite{#1}\_unskip}\_par
1593 \_ewref\_sxdef{ct!\_the\citenum}{\_string\mypage}%
1594 % \_vskip-.3\baselineskip
1595 \_endinsert
1596 }
1597 \_def\printCite#1{\_pdfliteral{2 Tr .15 w .9 g}#1\_pdfliteral{0 Tr 0 w 0 g}}
1598 \_def\printCite#1{{\_Grey#1}}
1599
1600 \_def\topinsertnopar{\_umidfalse \_upagefalse \_begingroup\_setbox0=\_vbox\_bgroup\_resetatrs}
```

op-bible.opm

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

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

```
1610 \_newbox\lqqbox
1611 \_newbox\rqqbox
1612 \_setbox\lqqbox=\_hbox{\_lower3pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed,}}
1613 \_setbox\rqqbox=\_hbox{\_kern2pt\_lower38pt\_hbox{\_setfontsize{at70pt}\_bf\LiRed"}}
1614 \_ht\lqqbox=0pt \_dp\lqqbox=0pt
1615 \_ht\rqqbox=0pt \_dp\rqqbox=0pt
1616
1617 \_def\quotedby{\_par}
1618 \_def\quotedbyright#1{%
1619 \_unskip\_nobreak\_hfill\_penalty0\_hskip2em
1620 \_null\_nobreak\_hskip\_iindent\_hbox{#1}}
```

op-bible.opm

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

```

1634 \_def\Cite #1#2{\_sdef{c!\_the\articlenum!#1}{#2}}
1635 \_def\insertCite #1#2{\_def\citelabel{#1}%
1636 \_ifx\_left#2\insertCiteleft
1637 \_else \_ifx#2\_right\insertCiteright\_else
1638 \_errmessage{\_noexpand\insertCite#1: \_noexpand\left or \_noexpand\right expected}%
1639 \_fi\_fi
1640 }
1641 \_def\insertCiteleft {%
1642 \_ifnum\citepg=1
1643 \_printwarn{\_noexpand\insertCite\citelabel: \_noexpand\swapCites activated}\_fi
1644 \_ifodd \_numexpr\_trys{cp!\_the\articlenum!\_citelabel}{0}+\_citepg\_relax
1645 \_else \_insertCitelr \_left \_fi
1646 }
1647 \_def\insertCiteright{%
1648 \_ifodd \_numexpr\_trys{cp!\_the\articlenum!\_citelabel}{0}+\_citepg\_relax
1649 \_insertCitelr \_right \_fi
1650 }
1651 \_def\insertCitelr#1{\_unskip\_vadjust{\_vbox{%
1652 \_ewref\_sxdef{{cp!\_the\articlenum!\_citelabel}{\_string\mypage}}}%
1653 \_vskip6pt
1654 \_advance\_hsize by\_parindent
1655 \_typosize[12/16]\_biGrey
1656 \_ifx#1\_left
1657 \_def\quotedby{\_par\_hfill}
1658 \_rightskip=\_parindent plus1fil \_leftskip=0pt
1659 \_setbox0\_vbox{%
1660 \_medskip \_noindent
1661 \_llap{\_copy\lqqbox}\_ignorespaces
1662 \_printCite{\_cs{c!\_the\articlenum!\_citelabel}}\_medskip}%
1663 \_hbox{\_kern-\_parindent\_rlap{White
1664 \_vrule height\_ht0 width\_hsize}\_box0}%
1665 \_else
1666 \_leftskip=\_parindent plus1fil
1667 \_parfillskip=0pt
1668 \_setbox0\_vbox{%
1669 \_medskip \_noindent
1670 \_rlap{\_hskip\_hsize\_kern-\_parindent\_copy\rqqbox}\_hfill
1671 \_ignorespaces \_printCite{\_cs{c!\_the\articlenum!\_citelabel}}\_medskip}%
1672 \_rlap{\_rlap{White \_vrule height\_ht0 width\_hsize}\_box0}%
1673 \_fi
1674 \_vskip6pt
1675 }}}
1676 \_def\swapCites{\_def\citepg{1}}
1677 \_def\citepg{0}
1678
1679 \_nspublic \Cite \insertCite ;

```

Insertions into the intro text

```

1687 %% TBN page 236
1688
1689 \_newcount\shapenum
1690 \_newdimen\ii \_newdimen\w
1691 \_def\oblom #1 od #2 odsadit #3 {\_par \_ii=#1 \_w=\_hsize
1692 \_ifdim\ii>\_zo \_advance\w by-\_ii
1693 \_else \_advance\w by\_ii \_ii=\_zo \_fi
1694 \_shapenum=1 \_tmpnum=0 \_def\shapelist{ }
1695 \_loop \_ifnum\shapenum<#2 \_edef\shapelist{\_shapelist\_zo\_hsize}%
1696 \_advance\shapenum by1 \_repeat
1697 \_loop \_edef\shapelist{\_shapelist\_ii\w}%
1698 \_advance\_tmpnum by1 \_ifnum\_tmpnum<#3 \_repeat
1699 \_advance\shapenum by#3 \_edef\shapelist{\_shapelist\_zo\_hsize}
1700 \_doshape}
1701 \_def\doshape{\_parshape \shapenum \shapelist}
1702 \_newcount\globpar
1703 \_ifx\_partokenset \_undefined \_def\partoken{\_par} \_else \_def\partoken{\_par} \_fi
1704 \_def\doshape{\_global\globpar=0 \_ea\_def\partoken{\_ifhmode\shapepar\_fi}}
1705 \_def\shapepar{\_prevgraf=\_globpar \_parshape\shapenum\shapelist
1706 \_endgraf \_global\globpar=\_prevgraf
1707 \_ifnum \_prevgraf>\shapenum \_ea\_let\partoken=\_endgraf \_fi

```

```

1708 }
1709
1710 \_def\Citehereleft #1 (#2) #3{{
1711   \_par
1712     \_def\quotedby{\_par\_hfill}
1713     \_rightskip=\_parindent plusfil \_leftskip=0pt
1714     \_setbox0\_vbox{{%
1715       \_typosize[12/16]\_bi\Grey
1716       \_hsize=.5\_hsize
1717       \_medskip \_noindent
1718       \_llap{\_copy\lqqbox}\_ignorespaces
1719       \_printCite{#3}\_medskip}}%
1720     \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1721     \_xdef\l\_lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1722     \_nointerlineskip\_vbox to0pt{\_kern#1\_baselineskip #2
1723       \_hbox{\_rlap{White
1724         \_kern-3mm\_vrule height\_ht0 width.5\_hsize}\_box0}}%
1725     \_vss}}
1726     \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1727     \_oblong {0.5\_tmpdim} od #1 odsadit {\_lines}
1728 }
1729 \_def\Citehereright #1 (#2) #3{{
1730   \_par
1731     \_def\quotedby{\_par\_parfillskip=0pt \_hfill}
1732     \_leftskip=\_parindent plusfill \_rightskip=0pt
1733     \_setbox0\_vbox{{%
1734       \_typosize[12/16]\_bi\Grey
1735       \_hsize=.5\_hsize
1736       \_vskip\_medskipamount \_rlap{\_kern\_hsize\_copy\lqqbox}\_vskip-\_medskipamount
1737       \_printCite{\_noindent\_ignorespaces#3}\_medskip}}%
1738     \_tmpdim=\_ht0 \_advance\_tmpdim by\_baselineskip
1739     \_xdef\l\_lines{\_the\_numexpr \_number\_tmpdim / \_number\_baselineskip \_relax}%
1740     \_nointerlineskip\_vbox to0pt{\_kern#1\_baselineskip #2
1741       \_hbox to\_hsize{\_hss
1742         \_llap{White \_vrule height\_ht0 width.5\_hsize \_kern-3mm}%
1743         \_llap{\_box0}}
1744     \_vss}}
1745     \_tmpdim=\_hsize \_advance\_tmpdim by-2\_leftskip
1746     \_oblong {-0.5\_tmpdim} od #1 odsadit {\_lines}
1747 }
1748
1749 \_def\Citehere{\_par \_ifodd\_pageno \_ea\Citehereright \_else \_ea\Citehereleft \_fi}
1750
1751 \_nspublic \Citehere ;

```

\insertBot {<title>} [<label>] (<params>) {<data>} inserts a material from <data> 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 <title> and it can be referred by <label>. The <params> can include a special setting used locally for the printing of this material.

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

op-bible.opm

```

1767 \_def\insertBot #1#2[#3]#4(#5)#6{% {Title} [label] (params) {data}
1768   \_botinsert
1769     \_leftskip=0pt \_rightskip=0pt \_relax
1770     \_botTitle{#1}[#3]%
1771     \_kern3pt \_nobreak
1772     \_vbox{\_picwidth=\_hsize #5 #6}%
1773   \_endbot
1774 }
1775 \_def\putBot #1 #2#3[#4]#5(#6)#7{% chap:verse {Title} [label] (params) {image-file.pdf}
1776   \_edef\fullvref{\_gentovref{#1}}%
1777   \_edef\fullvrefm{\_ea\renumvref\fullvref\_relax}%
1778   \_ea\newaction\_ea{\fullvrefm}{\insertBot{#2}[#4](#6){#7}}%
1779 }
1780 \_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`.

op-bible.opm

```
1789 \_def\.printintro{%
1790   \.begblock
1791     \_dest[i:\.currbook/]
1792     \_chaptit{\_mtext{intro}}}%
1793     \_input{\introfile}
1794   \.endblock
1795 }
```

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

op-bible.opm

```
1803 \_newcount\.blocklevel % nesting level of blocks
1804 \_def\.begblock{\_par\_bgroup
1805   \_advance\.blocklevel by1 \_advance\_leftskip by\_iindent \_rightskip=\_leftskip
1806   \_medskip
1807   \_pdfsavepos \_ea\_wref\_ea\Xblock\_ea{\_ea{\_the\.blocklevel}B{\_the\_pdflasttypos}}
1808   \_nobreak \_medskip
1809 }
1810 \_def\.endblock{\_par\_nobreak\_medskip
1811   \_pdfsavepos \_ea\_wref\_ea\Xblock\_ea{\_ea{\_the\.blocklevel}E{\_the\_pdflasttypos}}
1812   \_medskip \_egroup
1813 }
1814 \_refdecl{%
1815   \_def\Xblock#1#2#3{\_ifnum#1=1 \_edef\.tmp{frm:\_ea\_ignoresecond\_currpage}^^J
1816     \_unless\_ifcsize \_tmp \_endcsize \_sxdef{\_tmp}{\_fi^^J
1817     \_sxdef{\_tmp}{\_cs{\_tmp}#2#3}\_fi}
1818 }
1819 \_newdimen\_.frtop \_newdimen\_.frbottom % positions of top and bottom text on the pages
1820 \_def\_.frcolor{.93 g } % light grey -- color of blocks.
1821 \_pgbackground={%
1822   \_slet{\_opb\_tmp}{frm:\_the\_gpageno}
1823   \_ifx\_.tmp\_undefined \_def\_.tmp{\_fi
1824     \_.frtop=\_dimexpr \_pdfpageheight-\_voffset+\_smallskipamount\_relax
1825     \_.frbottom=\_dimexpr \_pdfpageheight-\_voffset-\_vsize-\_medskipamount\_relax
1826     \_ifx\_.frnext y \_edef\_.tmp{B{\_number\_.frtop}\_.tmp}\_global\_let\_.frnext n\_fi
1827     \_ea\_.printframes \_.tmp B{0}E{\_number\_.frbottom}
1828     \_ifx\_.frameslist\_empty \_else
1829     \_pdfliteral{q \_.frcolor 1 0 0 1 0 \_bp{-\_pdfpageheight} cm \_.frameslist Q}\_fi
1830   }
1831   \_def\_.printframes B#1#2E#3{\_ifnum#1=0 \_else
1832     \_.printframe {\_hoffset}{#3sp}{\_xhsize}{\_ifnum#1=-1 \_number\_.frtop\_else#1\_fi sp-#3sp}
1833     \_ifx^#2\_else \_global\_let\_.frnext=y \_let\_.printframes=\_relax \_fi
1834     \_ea\_.printframes\_fi
1835   }
1836   \_def\_.frameslist{}
1837   \_def\_.printframe #1#2#3#4{\_edef\_.frameslist{\_.frameslist
1838     \_bp{#1} \_bp{#2} \_bp{#3} \_bp{#4} re f }%
1839 }
```

Insertions objects over pictures (maps)

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

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

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

op-bible.opm

```
1860 \_def\_.putstext{\_ea\_ea\_ea\_.putstextA\_scantwodimens}
1861 \_def\_.putstextA#1#2#3{%
1862   \_setbox0=\_hbox{\_.shadowedtext{#3}}%
1863   \_dimen1=#1sp \_dimen2=#2sp \_puttextB
1864 }
1865 \_def\_.shadowedtext#1{%
1866   \_insertwhiteshadowresources
```

```

1867 \_setbox0=\_hbox{#1}%
1868 \_hbox{\_tmpdim=\_ht0 \_advance\_tmpdim by\_dp0
1869 \_lower\_dp0\_hbox{
1870 \_pdflliteral{q /trans gs 1 g
1871 \_for num 1..10\_do{\_oval{\_bp{\_wd0}}{\_bp{\_tmpdim}}{2+##1/2} f } Q}}%
1872 \_box0}%
1873 }
1874 \_def\insertwhiteshadowresources{
1875 \_addextgstate{trans}{<</ca \shadowparameter>>}%
1876 \_glet\insertwhiteshadowresources=\_relax
1877 }
1878 \def\shadowparameter{.1} % default value of "transparency"
1879
1880 \_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.

op-bible.opm

```

1889 \_def\c[#1/#2]#3{% text podel krivky: \c[init-rotace/repetice]{text}
1890 \_pdfsave\_pdfrotate{#1}\_rlap{\_edef\_.tmpb{#3}\_replstring\_.tmpb{ }{ } \_def\_.tmpa{#2}%
1891 \_ea\_foreach\_.tmpb\_do{##1\_.tmpa}}\_pdfrestore \_kern10mm
1892 }
1893 \_let\c=\_undefined
1894 \_nspublic \c ;

```

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

op-bible.opm

```

1902 \_def\townparams{ % default parameters of the circle:
1903 \_hhkern=.8pt % diameter of the disc
1904 \_lwidth=.5pt % tickness of the outline
1905 \_fcolor=\Red % color of the inner disc
1906 \_lcolor=\Black % color of the outline
1907 }
1908 \_def\town {\_ea\_ea\_ea\_.townA\_scantwodimens}
1909 \_def\townA #1#2{\_setbox0=\_hbox{\_incircle[\_hhkern=0pt \_vvkern=0pt \townparams]{} }%
1910 \_dimen1=#1sp \_dimen2=#2sp \_puttextB
1911 }
1912 \_nspublic \town ;

```

14 Chiasm

The pair `\begChiasm... \endChiasm` defines chiasm environment. 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

```

op-bible.opm

```

1935 \_def\keepstyle{\_defaultitem=\_printitem}
1936 \_def\easylist{\_adeft*\countlist}
1937 \_def\ast{\countlist}
1938 \_def\countlist{\_tmpnum=1 \countlistA}
1939 \_def\countlistA{\_futurelet\_.next\countlistB}
1940 \_def\countlistB{\_ifx\_.next\ast \_ea\countlistC\_else \_ea\countlistD \_fi}
1941 \_def\countlistC#1{\_incr\_tmpnum \countlistA}
1942 \_def\countlistD{%
1943 \_ifnum\_tmpnum>\_ilevel \_for num \_ilevel..\_tmpnum-1 \_do{\begitems\easylist}\_else
1944 \_ifnum\_tmpnum<\_ilevel \_for num \_tmpnum..\_ilevel-1 \_do{\enditems}\_fi\_fi

```

```

1945 \_startitem}
1946
1947 \_def\qq#1{{\bf#1\_trycs{Level:\_the\_ilevel}{}}\_space\_aftergroup\qqA}
1948 \_def\qqA{\_sdef{Level:\_the\_ilevel}{\_rlap{'}}}
1949 \_def\ChiasmNumbering{\_ea\qq \_Uchar \_numexpr `A-1+\_ilevel\_relax\_space} % A, B, C, D, etc.
1950 \_sdef{\_item:q}{}%for chiasms with no leading alphabet letters
1951 \_sdef{\_item:Q}{\ChiasmNumbering}
1952 \_def\beginChiasm{\_begitems \_easylist \_style Q \_keepstyle}
1953 \_def\endChiasm{\_for num 1..\_ilevel \_do{\_enditems}}
1954
1955 \_nspublic \beginChiasm \endChiasm ;

```

15 Outline

op-bible.opm

```

1963 \_newdimen\colsep
1964 \colsep=10pt
1965
1966 \_def\Outline{
1967 \_medskip
1968 % \filbreak
1969 \_chaptit{\_mtext{outline}}}%
1970 \_everylist={\_ifcase\_ilevel \_or \_style I \_or \_style A \_or \_style n \_fi}
1971 \_sdef{\_item:A}{\_strut\_uppercase\_ea{\_athe\_itemnum}. }
1972 \_sdef{\_item:I}{\_strut\_uppercase\_ea{\_romannumeral\_itemnum}. }
1973 \_hsize=.5\_hsize \_advance\_hsize by-\colsep
1974 \_emergencystretch=40pt
1975 \_leftskip=0pt \_rightskip=0pt
1976 }
1977 \_def\rightnote#1{\_par
1978 \_setbox0=\_hbox{\_kern\_hsize \_kern\colsep
1979 \_vtop{\_leftskip=0pt \_kern0pt\_noindent\_strut\_it#1}}
1980 \_ht0=0pt \_dp0=0pt \_box0 \_nointerlineskip
1981 }
1982 \_nspublic \Outline \rightnote ;

```

16 Timelines

- `\timeline<num>` sets the totla 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).

op-bible.opm

```

1995 \_def\l{\_baselineskip}
1996 \_newcount\.\timeline \.\timeline=100 % default
1997 \_newdimen\.\tlwidth \.\tlwidth=10cm % default
1998 \_def\.\timelinewidth{\_afterassignment\.\timelinewidthA\.\tlwidth}
1999 \_def\.\timelinewidthA{\_par\_hbox to\.\tlwidth{}}
2000
2001 \_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.

op-bible.opm

```

2014 \_def\.\arrowtext #1..#2(#3)#4{%
2015 \_puttext \.\pos{#1}0pt
2016 {\_lower.745ex\_hbox to\_dimexpr\.\pos{#2}-\.\pos{#1}{#3}\_Larrow{ #4 }\_Rarrow}}
2017 }
2018 \_def\.\Larrow{$_\leftarrow$_\_kern-.8em\_leaders\_vrule height.65ex depth-.42ex\_hfil}
2019 \_def\.\Rarrow{\_leaders\_vrule height.65ex depth-.42ex\_hfil\_kern-.8em$_\rightarrow$}
2020 \_def\.\rule{\_leaders\_vrule height.12ex depth.12ex\_hfil}
2021 \_def\.\pos#1{\_expr{#1/\_the\.\timeline}\.\tlwidth}
2022
2023 \_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, .If nothing is here the text is centered. The *<format of text>* can include the font setting, color setting etc.

op-bible.opm

```
2036 \_def\tlput #1 #2 #3(#4)#5{%
2037   \_let\Lhss=\_hss \_let\Rhss=\_hss
2038   \_ifx#3\_rlap\_relax \_let\Lhss=\_relax \_let\Rhss=\_hss \_fi
2039   \_ifx#3\_llap\_relax \_let\Lhss=\_hss \_let\Rhss=\_relax \_fi
2040   \_puttext \_pos{#2}Opt {\_hbox toOpt{\_Lhss #4\tltext#1{#5}\_Rhss}}
2041 }
2042 \_def\tltext#1#2{\_ifx#1a\_vbox\_else
2043   \_vtop\_fi{\_kern0pt\_halign{\_Lhss#\_Rhss\_cr\_strut#2\_crcr}}}%
2044 }
2045 \_nspublic \tlput ;
```

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

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

op-bible.opm

```
2055 \_def\tline #1..#2 {%
2056   \_puttext \_pos{#1}Opt {\_hbox to \_dimexpr\_pos{#2}-\_pos{#1}{\_rule}}
2057 }
2058 \_def\tlines#1{\_puttext OptOpt{\_hbox{\_foreach #1|\_do#1|{\_vrul\_hskip\_pos{0##1}}}}
2059 \_def\vrul{\_def\vrul{\_kern-.12ex\_vrule height.7\.1 depth.7\.1 width.24ex \_kern-.12ex}}
2060
2061 \_nspublic \tline \tlines ;
```

17 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, cquotes characters are removed and outer margins are reduced because there is no material in them.

op-bible.opm

```
2075 \_def\normalchapnumbers{
2076   \_margins/2 a4 (25,25,20,20)mm
2077   \_lrmargin=0pt
2078   \_setbox0=\_box\lqqbox \_setbox0=\_box\rqqbox
2079   \_def\printbeforefirst{%
2080     \_nobreak\_medskip
2081     \_trychapnote
2082     \_hangindent=\_parindent \_hangafter=-2
2083     \_noindent \_llap{\_vbox toOpt
2084       {\_kern-8pt\_hbox{\_setfontsize{at23pt}\_bf\Red\_the\chapnum\_kern5pt}\_vss}}%
2085   }
2086 }
2087 \_nspublic \normalchapnumbers ;
```

18 Checking syntax

op-bible.opm

```
2095 \_def\checksyntax#1 {%
2096   \_let\processbooks=\_relax
2097   \_ifx\_relax#1\_relax \_else
2098     \_begingroup
2099       \_the\syntaxmacros
2100       \_wterm{^^J** checking file: #1 **^^J}
2101       \_input{#1}
2102       \_vfil\_break
2103     \_endgroup
2104     \_ea\checksyntax \_fi
2105 }
2106
2107 \_newtoks\syntaxmacros
2108 {\_catcode`<=13
```

```

2109 \_global\syntaxmacros={
2110 \_def<#1>{\_bgroup
2111 \_message{checking \_unexpanded{<#1>}}%
2112 \_ifx\_relax#1\_relax \_errmessage{empty link}\_nobref\_else \_afterfi{\_checkbref#1>\_bref#1>}\_fi
2113 \_glet\linkpre=\linkpre \_glet\linkspec=\linkspec
2114 \_egroup
2115 }
2116 \_def\checkbref#1#2>{%
2117 \_isinlist{.#1#2}{<}\_iftrue \_errmessage{duplicated \_string<}\_nobref\_else
2118 \_ifx"#1\checkbrefQ #1#2>\_else \_checkbrefD #1#2>\_fi\_fi
2119 }
2120 \_def\checkbrefQ "#1"#2#3>{\_checkbrefD #2#3>}
2121 \_def\checkbrefD #1>{%
2122 \_isinlist{.#1}{ }\_iftrue\checkbrefS#1>\_else\checkbrefN#1>\_fi
2123 }
2124 \_def\checkbrefS #1 #2>{\_checkbrefN#2>}
2125 \_def\checkbrefN #1>{%
2126 \_def\tpmb{#1}
2127 \_ifx\tpmb\_empty \_errmessage{missing link data}\_nobref\_else
2128 \_replstring\tpmb{:}{ }\_replstring\tpmb{-}{ }\_replstring\tpmb{_{ }}%
2129 \_replstring\tpmb{a}{ }\_replstring\tpmb{b}{ }\_replstring\tpmb{c}{ }\%
2130 \_setbox0=\hbox{\_tmpnum=0\tpmb\_relax}%
2131 \_ifdim\_wd0>0pt \_errmessage{nonnumeric link data}\_nobref\_fi
2132 \_fi
2133 }
2134 \_def\_nobref{\_def\_bref##1>{\_Red\_string<##1>}}
2135 \_def\currbook{}
2136 \_def\prelinkB{BK}
2137 \_def\prelinkC{BK}
2138 \_def\prelinkV{0}
2139 \_def\nochapbooks{BK}
2140 \_let<=<
2141
2142 \_def\x/#1/{\_def\tpmb{#1}%
2143 \_isinlist\tpmb\x\_iftrue \_badx
2144 \_else \_isinlist\tpmb<\_iftrue \_badx
2145 \_else \_isinlist\tpmb\enditems\_iftrue \_badx \_else \_x/#1/\_fi\_fi\_fi
2146 }
2147 \_def\_badx{\_errmessage{unclosed \_string\x/.../}}
2148
2149 \_def\Article[#1]{ }
2150 \_def\Cite #1 {\_par\_noindent{\_bf Cite: }}
2151 \_def\insertCite #1#2{}
2152
2153 \_def\putArticle #1 #2[#3]#4(#5){ }
2154 \_def\putCite #1:#2 {\_par\_noindent{\_bf Cite: }}
2155 \_def\putBot #1 #2[#3]#4(#5){\_vbox}
2156
2157 \_def\c[#1/#2]#3{#3}
2158
2159 \_long\_ea\_def\_csname Note\_endcsname #1 #2#3%
2160
2161 {\_par \_let\nextww\_undefined \_noindent{\_bf Note #1:} #3\_par}
2162 }}
2163 \_nspublic \checksyntax ;

```

19 TODO macros

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

op-bible.opm

```

2173
2174 \_def\quotationmarks#1#2{%
2175 \_cnvtext{"}{\_doquotation}%
2176 \_def\doquotation {\_futurelet\next\_doquotationA}%
2177 \_def\doquotationA {%
2178 \_let\doquotationB=#1\_relax
2179 \_ea\_ifx\_space\next \_let\doquotationB=#2\_fi
2180 \_ifx\_space\next \_let\doquotationB=#2\_fi

```

```

2181 \_ifx\_endgraf\.next \_let\doquotmarkB=#2\_fi
2182 \_ifx\_endcenter\.next \_let\doquotmarkB=#2\_fi
2183 \_ifx.\.next \_let\doquotmarkB=#2\_fi
2184 \_ifx,\.next \_let\doquotmarkB=#2\_fi
2185 \doquotmarkB}%
2186 }
2187 \_nspublic \quotationmarks ;
2188
2189 \_def\chaptit#1{\_line{\_hss\chapfont\Red#1\_hss}
2190 \_nobreak
2191 }
2192 \_def\schaptit#1{\_bigskip\chaptit{#1}\_nobreak\_medskip}
2193
2194 \_def\subtit#1{\_par
2195 \_ifnum\currversenum=1 \_else \_medskip\_fi
2196 \_line{\_indent\subtitfont #1\_hss}\_nobreak
2197 \_ifnum\currversenum=1 \_vskip-\_medskipamount\_fi
2198 \_smallskip
2199 }
2200 \_def\subtitfont {\Red\_it}
2201
2202 \_nspublic \chaptit \schaptit \subtit ;
2203
2204 \_sdef\mt:intro:en{Introduction} \_sdef\mt:outline:en{Outline}
2205 \_sdef\mt:intro:cs{Úvod} \_sdef\mt:outline:cs{Osnova}
2206
2207 \_def\dopsat{{\Red !!! DOPSAT !!! }}
2208
2209 \_def\bibleinput#1 {\_bgroup
2210 \_catcode`##=13 \_bgroup\_lccode`~=`## \_lowercase{\_egroup\_let~}=\_processline
2211 \_input{#1}%
2212 \_egroup
2213 }
2214 \_let\FormattedBook=\_ignoreit % for backward compatibility
2215 \_let\CommentedBook=\_ignoreit % for backward compatibility

```

Active character < used for references.

op-bible.opm

```

2221 \_outer\_def\Note {\_Note}
2222 \_outer\_def\ww {\_ww}
2223 \_outer\_def\ChapterPre {\_ChapterPre}
2224 \_outer\_def\ChapterPost {\_ChapterPost}
2225
2226 \_def\_afterload{\_adef<{\_bref}}
2227 \_afterload
2228
2229 \_endnamespace

```

20 Index

\. 2–9, 11–14, 16–18, 23	\BookTile 3	\chapbefore 10
\.AddNote 5	\.botins 17	\checknochapbooks 3
\.addpre 8	\.botinsert 17	\Cnotetext 7
\alist! 4, 9	\.botTitle 17	\cnvtext 9
\amark 2	\bpo! 2, 4	\createlink 13
\Article 17	\bpr! 2, 4	\currbook 2
\.begblock 23	\.bref 10	\currchapnum 9
\begcenter 8	\.brefBookChapter 3	\currverse 9
\begChiasm 24	\.brefL 12	\currversenum 9
\bex! 2–3	\.btit 2	\currversetext 9
\bibname 3	\btit! 3	\doArticle 17
\bmark 2–3	\.buff 8–9	\doCNote 7
\BookException 2–3	\c 24	\doImage 17
\BookPost 2, 4	\centeringmode 8	\doNote 7
\BookPre 2, 4	\chapafter 10	\endblock 23

<code>\.endbot</code> 17	<code>\megrednotes</code> 7	<code>\putBot</code> 22
<code>\endcenter</code> 8	<code>\.myaddto</code> 1	<code>\putImage</code> 17
<code>\endChiasm</code> 24	<code>\.newaction</code> 4	<code>\putstext</code> 23
<code>\.ensuredest</code> 13	<code>\.newbook</code> 3	<code>\pword!</code> 5
<code>\everybref</code> 12	<code>\.newlinkB</code> 12	<code>\reduceref</code> 12
<code>\f!</code> 3	<code>\.nextww</code> 14	<code>\renum</code> 5, 11–12, 15
<code>\fmtfile</code> 2	<code>\.nextwwA</code> 14	<code>\renumlabel</code> 6
<code>\fmtins</code> 4, 8	<code>\nochapbooks</code> 3	<code>\renumlinktext</code> 12
<code>\.fmtpoetA</code> 8	<code>\Note</code> 4–8, 14–15	<code>\renumvref</code> 5
<code>\.fmtpoetB</code> 8	<code>\.NoteB</code> 6	<code>\replpre</code> 4
<code>\.fmtpoetC</code> 8	<code>\.notefail</code> 6	<code>\replprepost</code> 4
<code>\fmtpoetry</code> 8, 10	<code>\.noteins</code> 16	<code>\sedef</code> 1
<code>\fmtpre</code> 8, 10	<code>\.noteinsert</code> 16	<code>\.setheadline</code> 3
<code>\.fmtprebuf</code> 10	<code>\.notelog</code> 8	<code>\.setvarnum</code> 15
<code>\.fmtprebuff</code> 8	<code>\.notenunum</code> 5	<code>\shadowedtext</code> 23
<code>\fmtprepoet</code> 8	<code>\notepre!</code> 5, 7	<code>\shadowparameter</code> 23
<code>\ftmadd</code> 8	<code>\noteref!</code> 5	<code>\switch</code> 15
<code>\.fullvref</code> 5	<code>\noterule</code> 16	<code>\.switchA</code> 15
<code>\.fullvrefm</code> 5	<code>\.noteset</code> 16	<code>\.switchD</code> 15
<code>\.gentovref</code> 5	<code>\notesfile</code> 2	<code>\.switchN</code> 15
<code>\.hboxorllap</code> 10	<code>\notetext!</code> 5	<code>\timeline</code> 25
<code>\ind</code> 8, 10	<code>\notracinglinks</code> 12–13	<code>\timelinewidth</code> 25
<code>\insertBot</code> 22	<code>\.numvariants</code> 13	<code>\tline</code> 26
<code>\insertSpanImage</code> 18	<code>\.oncebref</code> 12	<code>\tlines</code> 26
<code>\insertSpanText</code> 19	<code>\pbook!</code> 2	<code>\tmark</code> 14–15
<code>\introfile</code> 2, 23	<code>\pg</code> 13	<code>\town</code> 24
<code>\.iscolonin</code> 2	<code>\.prebuff</code> 7	<code>\townparams</code> 24
<code>\.isdivisin</code> 2	<code>\.prelinkB</code> 12	<code>\tracinglinks</code> 13
<code>\.isspacein</code> 2	<code>\.prelinkC</code> 12	<code>\.transformword</code> 5
<code>\l</code> 25	<code>\.prevnotepre</code> 7	<code>\.trychapnote</code> 10
<code>\.linkfspec</code> 11	<code>\.printbeforefirst</code> 10	<code>\.upcasefirst</code> 7
<code>\.linkfspecone</code> 12	<code>\.printchapnote</code> 10	<code>\v!</code> 14
<code>\.linklog</code> 12	<code>\.printCnote</code> 7	<code>\variants</code> 13, 15
<code>\.linkpre</code> 11	<code>\printedbooks</code> 2	<code>\varnum</code> 15
<code>\.linktext</code> 11	<code>\.printintro</code> 23	<code>\vdef</code> 14
<code>\.ltextB</code> 10	<code>\.printverse</code> 9	<code>\.vdefB</code> 14
<code>\.ltextC</code> 10	<code>\.printwarn</code> 1	<code>\.versedef</code> 11
<code>\.ltextF</code> 11	<code>\processbooks</code> 2–3	<code>\ww</code> 5, 14–15
<code>\.ltextN</code> 11	<code>\.processline</code> 9	<code>\x</code> 14
<code>\.ltextP</code> 10	<code>\.processverse</code> 9	<code>\xA</code> 14
<code>\.ltextS</code> 11	<code>\.punctpword</code> 7	<code>\.Xdest</code> 13
<code>\.ltextV</code> 10	<code>\putArticle</code> 17	