Basic settings:

op-bible.opm

```
\load[vlna]
\load[mte]

\isfile{f-biblon.opm}\iftrue
   \fontfam[biblon]
\else
   \fontfam[lmfonts]
\fi

\chyph
\enablemte

\typosize[11/13]
\hyperlinks\Blue\Blue

\raggedbottom
\parindent=20pt
```

Fonts:

op-bible.opm

```
\fontdef\bookfont{\setfontsize{at19.pt}\bf}
\fontdef\chapfont{\setfontsize{at13.pt}\bf}
\fontdef\markfont{\setfontsize{at7pt}\rm}
```

⟨*full-vref*⟩ is full reference to verse in the format ⟨*book-mark*⟩/⟨*chapter-num*⟩:⟨*verse-num*⟩
`\alist!`⟨*full-vref*⟩ expands to the list of replace action numbers applied for given verse. The actions are referenced by its number. Each new action has newly allocated action-number.
`\action!`⟨*number*⟩ expands to the replace action, we suppose that the corresponding verse in saved to `\tmpb`
`\newaction{`⟨*full-vref*⟩`}{`⟨*action-body*⟩`}` allocates new action.

op-bible.opm

```
\newcount\actionnum
\def\newaction#1#2{\incr\actionnum
   \unless\ifcsname alist!#1\endcsname \sxdef{alist!#1}{}\fi
   \sxdef{alist!#1}{\cs{alist!#1}\the\actionnum,}%
   \global\sdef{action!\the\actionnum}{#2}%
}
```

`\replpre{`⟨*prefix*⟩`}{`⟨*text*⟩`}{`⟨*fail*⟩`}` replaces first occurence of ⟨*text*⟩ by ⟨*prefix*⟩`{`⟨*text*⟩`}` in `\tmpb` macro. If the ⟨*text*⟩ is empty then ⟨*prefix*⟩`{}` is inserted at the beginning of the `\tmpb`.
If ⟨*text*⟩ does not exists then ⟨*fail*⟩ is processed. The ⟨*fail*⟩ macro can use `\text` where ⟨*text*⟩ is saved.

op-bible.opm

```
\def\replpre#1#2#3{%
   \ifx^#2^\def\tmp{#1{}}\ea\ea\ea\def\ea\ea\ea\tmpb\ea\ea\ea{\ea\tmp\tmpb}%
   \else
      \def\replpredo##1#2##2\end{%
         \ifx^##2^\def\text{#2}#3% <fail>
         \else \replsave ##1#1{#2}##2\end \fi
      }%
      \def\replsave##1#2\end{\def\tmpb{##1}}%
      \ea\replpredo\tmpb#2\end
   \fi
}
```

The ⟨*gen-vref*⟩ is generalized reference to the verse. It can be ⟨*chapter-num*⟩`:`⟨*verse*⟩ (the ⟨*book-mark*⟩ is appended from `\CommentedBook` token list) or ⟨*chapter-num*⟩`:`⟨*verse-from*⟩`-`⟨*verse-to*⟩ (only ⟨*verse-from*⟩ is used for generating ⟨*gen-vref*⟩.
`\gentovref{`⟨*gen-vref*⟩`}` expands to ⟨*full-vref*⟩.

op-bible.opm

```
\newtoks\CommentedBook
\def\gentovref#1{\the\CommentedBook/\gentovrefA#1-\end}
\def\gentovrefA#1-#2\end{#1}
```

`\renumref` ⟨*full-vref*⟩`\relax` does re-caclulating of ⟨*full-vref*⟩ using `\renum` data.

op-bible.opm

```
\def\renumvref #1:#2\relax{#1:\trycs{rn!\tmark!#1:#2}{#2}}
```

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

op-bible.opm

```
\def\transformword#1{%
   \ifcsname w!\fullvref!\tmark!#1\endcsname \lastnamedcs
   \else \ifcsname v!\tmark!#1\endcsname \lastnamedcs
   \else #1\fi\fi
}
```

`\Note` ⟨*gen-vref*⟩ ⟨*space*⟩ `{`⟨*word*⟩`}` ⟨*text*⟩ ⟨*empty-line*⟩ transforms ⟨*word*⟩ to the ⟨*tword*⟩ (see above), saves ⟨*text*⟩ and activates replace-action of ⟨*tword*⟩ to `\doNote{`⟨*note-num*⟩`}{`⟨*tword*⟩`}` in given verse.
`\Note` does following:

- Allocates new ⟨*note-num*⟩,
- Transforms ⟨*gen-vref*⟩ to ⟨*full-vref*⟩ using `\gentovref`.
- Transforms ⟨*word*⟩ to ⟨*tword*⟩ (to be searched and printed) by `\transformword`.
- Reads ⟨*pword*⟩ (word to be printed in the note) if alternative syntax with `={`⟨*pword*⟩`}` is used. Else ⟨*pword*⟩ is equal to ⟨*tword*⟩.
- Defines `\notetext!`⟨*note-num*⟩ as ⟨*text*⟩.
- Defines `\noteref!`⟨*note-num*⟩ as ⟨*full-vref*⟩.
- Defines `\notepre!`⟨*note-num*⟩ as ⟨*gen-vref*⟩.
- Defines `\pword!`⟨*note-num*⟩ as ⟨*pword*⟩,
- Does `\newaction{`⟨*full-vref*⟩`}{\replpre{\doNote{`⟨*note-num*⟩`}}{`⟨*tword*⟩`}{\notefail{`⟨*note-num*⟩`}}}`.

The `\Note` macro has an alternative syntax

`\Note` ⟨*gen-vref*⟩ ⟨*space*⟩ `{`⟨*word*⟩`}={`⟨*pword*⟩`}` ⟨*text*⟩ ⟨*empty-line*⟩

If ⟨*pword*⟩ is given then is is printed in the note instead ⟨*tword*⟩. More preciselly: transformed ⟨*word*⟩ is used for searching (and it is kept in the verse unchanged) but ⟨*pword*⟩ is printed in the note.

op-bible.opm

```
\newcount\notenum
\outer\def\Note #1 #2{%
   \incr\notenum
   \sxdef{notepre!\the\notenum}{#1}%
   \edef\fullvref{\gentovref{#1}}%
   \edef\fullvrefm{\ea\renumvref\fullvref\relax}%
   \edef\tword{\transformword{#2}}%
   \edef\oword{#2}%
   \isnextchar={\NoteA}{\NoteA={}}%
}
\def\NoteA=#1 #2\par{%
   \global\sdef{notetext!\the\notenum}{#2}%
   \sxdef{noteref!\the\notenum}{\fullvrefm}%
   \ifx^#1^\sxdef{pword!\the\notenum}{\tword}\fi
   \ifcsname ww!\fullvref!\tmark!\oword \endcsname \global\slet{pword!\the\notenum}{}\fi
   \ifx^#1^\else \global\sdef{pword!\the\notenum}{#1}\fi
   \edef\tmp{%
      \noexpand\newaction{\fullvrefm}%
       {\noexpand\replpre{\noexpand\doNote{\the\notenum}}{\tword}{\noexpand\notefail{\the\notenum}}}}%
   \tmp
}
\def\notefail#1{%
   \printwarn{\csstring\\Note: \currverse: The text "\unexpanded\ea{\text}" not found}%
   \replpre{\doNote{#1}}{}{}% \Note is registered with the beginning of the verse
}
\def\printwarn#1{\wterm{WARNING (l.\the\inputlineno) #1}}
```

When bible-text (from sword) 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 `\processverse` ⟨*full-vref*⟩⟨*space*⟩⟨*verse-text*⟩`\end` is repeatedly processed.

```
\eoldef\processline#1{\processverse \currbook/#1\end}
```

`\processverse` ⟨*full-vref*⟩⟨*space*⟩⟨*verse-text*⟩`\end` does

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


```
\newcount\chapnum
\def\processverse #1 #2\end{%
   \edef\currverse{#1}%
   \preparechapverse #1
   \def\tmpb{#2}\def\tmpc{}%
   \ifcsname alist!#1\endcsname \ea\ea\ea\processactions \csname alist!#1\endcsname 0,\fi
   \ifnum\currchapnum=\chapnum \else \chapnum=\currchapnum\relax \printchap \fi
   \printverse
}
\def\processactions #1,{\ifnum #1=0
   \else \cs{action!#1}%
   \ea \processactions \fi
}
\def\preparechapverse #1/#2:#3 {\def\currchapnum{#2}\def\currversenum{#3}}
```

`\printverse` prints verse from `\currversenum` and (possibly changed) `\tmpb`. It prints the single raised verse number first.
`\printchap` prints beginning od new chapter. `\printbeforefirst` is a macro which is executed just before first verse of the chapter, after all material from `\fmtpre` is executed. I.e after printing a chatper name (if declered by `\fmtpre`).


```
\def\printverse{%
   \tmpc % material accumulated by \fmtpre
   \ifnum\currversenum=1 \printbeforefirst \fi
   \quitvmode \trymakedest{v:\currverse}%
   \raise5pt\hbox{\unless\ifnum\currversenum=1 \markfont\currversenum\fi}%
   \tmpb \space
}
\def\printchap{\bigskip}
%   \par\removelastskip
%   \medskip
%   {\chapfont\Red \the\chapnum}\par\nobreak\medskip
%}
\def\printbeforefirst{%
   \par\nobreak
   \vbox to0pt{\null\vskip-1ex
      \hbox to\parindent{\hss \chapfont\Red \the\chapnum\ \hss}\vss}\nobreak \vskip-2ex
   \noindent \hangindent=\parindent \hangafter=-2 \relax}
```

`\fmtpre {`⟨*gen-vref*⟩`}{`⟨*what*⟩`}` adds ⟨*what*⟩ to `\tmpc`, i.e. at the beginning of the verse.
`\ftmadd {`⟨*gen-vref*⟩`}{`⟨*what*⟩`}` adds ⟨*what*⟩ to `\tmpb`, 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 th ⟨*what*⟩ is inserted like `\fmtpre` does it
All these commands allocate new action using `\newaction`.


```
\let\FormatedBook=\CommentedBook
\def\fmtpre#1#2{\newaction{\gentovref{#1}}{\addto\tmpc{#2}}}
\def\fmtadd#1#2{\newaction{\gentovref{#1}}{\addto\tmpb{#2}}}
\def\fmtins#1#2#3{\newaction{\gentovref{#1}}{\replpre{\fmtafter{#3}}{#2}{\fmtfail{#3}}}}
\def\fmtafter#1#2{#2#1}
\def\fmtfail#1{\fmtwarn\addto\tmpc{#1}}
\def\fmtwarn{\printwarn{\string\fmtins: \currverse: The text "\unexpanded\ea{\text}" not found}}
```

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

op-bible.opm

```
\newdimen\centermargin \centermargin=4em
\def\begcenter{\par \medskip
   \bgroup
   \def\centeringmode{y}
   \parindent=0pt
   \leftskip=\centermargin plus1fill
   \rightskip=\leftskip
}
\def\endcenter{\par
   \ifx\centeringmode\undefined
      \printwarn{\noexpand\endcenter ignored: no \noexpand\begcenter precedes}
   \else \egroup \medskip \fi}

%% Makra pro tvorbu linků specifikovaných jako <link>
```

`\bref` ⟨*link-spec*⟩`>` creates a ling given by ⟨*link-spec*⟩ and prints ⟨*link-spec*⟩. See user manual for variants of the format of ⟨*link-spec*⟩. We will set `<` as active character with meaning `\bref`, so user can type `<`⟨*link-spec*⟩`>` for link specification.
Internal full link specification is ⟨*link-pre*⟩:⟨*book-mark*⟩/⟨*chapter-num*⟩:⟨*verse-num*⟩. The macro `\bref` reads ⟨*link-spec*⟩ and creates internal full link specification saved in `\linkpe:\linkfspec` macros. Moreover, it creates macro `\linktext` wich expands to the text to be printed as active link.

op-bible.opm

```
\def\bref {\futurelet\next\brefC}
```

Macro `\brefC` scans `"` as a potential first character and saves ⟨*word*⟩ (from `"`⟨*word*⟩`"` specification) to `\linktext`. If there isn't `"` then `\linktext` is set as empty macro. The next specification will be addet to the `\linktex` later.

op-bible.opm

```
\def\brefB {\futurelet\next\brefC}
\def\brefC {\ifx\next"\def\nextdo"##1"{\def\linktext{##1}\brefD}%
            \else \def\linktext{}\def\nextdo{\brefD}%
            \fi \nextdo
}
```

If there si a space between `"`⟨*word*⟩`"` and the next ⟨*link-spec*⟩, then it is saved as no-breakable space into `\linktext`.

op-bible.opm

```
\def\brefD {\futurelet\next\brefE}
\def\brefE {\ea\ifx\space\next \addto\linktext{~}\def\nextdo{\afterassignment\brefF \let\next= }%
            \else \def\nextdo{\brefF}%
            \fi \nextdo
}
```

Next ⟨*link-spec*⟩ is processed by macros `\brefF` ... `\brefJ`.

op-bible.opm

```
\def\brefF #1>{\brefG #1 >}
\def\brefG #1 #2>{\ifx^#2^\edef\linkfspec{\currbook/}\afterfi{\brefH #1:>}%
                  \else \addto\linktext{#1~}\def\linkfspec{#1/}\afterfi{\brefGH#2:>}%
                  \fi
}
\def\brefGH #1 {\brefH #1}
\def\brefH #1:#2>{\ifx^#2^\ea\addto\ea\linkfspec\ea{\the\chapnum:}\afterfi{\brefI #1->}%
                  \else \addto\linktext{#1:}\addto\linkfspec{#1:}\afterfi{\brefHI #2->}%
                  \fi
}
\def\brefHI #1:{\brefI #1}

\def\brefI #1-#2>{\addto\linktext{#1}\addto\linkfspec{#1}%
                  \ifx^#2^\afterfi{\futurelet\next\brefK}%
                  \else \afterfi{\brefJ#2}\fi
}
\def\brefJ#1-{\addto\linktext{--#1}\futurelet\next\brefK}

\def\brefK{\def\linkpre{v}% default
   \ea\striptocomma\linkfspec,\end\linkfspec
   \ifx n\next \def\linkpre{n}\ea\brefKK\fi
   \brefL
}
\def\brefKK #1\brefL#2{\brefL}

\def\striptocomma #1,#2\end#3{\def#3{#1}}
```

Macro `\brefL` creates the link `\linkpre:\linkfspec` with the text `\linktext`.

op-bible.opm

```
\def\brefL{\ensuredest \createlink}
```

`\createlink` creates link only if it refeneces 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`

op-bible.opm

```
\def\createlink{\ea\isprintedbook\linkfspec \iftrue
   \link[\linkpre:\linkfspec]{\Blue}{\linktext}%
   \else {\Blue\linktext}\fi
}
\def\isprintedbook #1/#2\iftrue{\ifcsname pbook!#1\endcsname}
```

We don't create destinations for all verses, notes etc. but only for those which are referenced. Macro `\ensuredest` creates the item `\Xcreatedest` to .ref file and it is read in the second TEX run. The `\trymakedest` macro is used ad the begining of each verse, note etc. Only referenced destinations are created.

op-bible.opm

```
\def\ensuredest{\openref \immediate\_wref\Xcreatedest{{\linkpre:\linkfspec}}}
\refdecl{
   \def\Xcreatedest#1{\sxdef{dest!#1}{}}
}
\def\trymakedest#1{\ifcsname dest!#1\endcsname \dest[#1]%
   \global \ea\let\csname dest!#1\endcsname \undefined \fi}
```

The macros `\BookTile` ⟨*b-mark*⟩ ⟨*a-mark*⟩ `{`⟨*title*⟩`}` declare titles of each Bible books. The ⟨*b-mark*⟩ is a book mark used in file names and ⟨*a-mark*⟩ is an actual book mark used in printed text.

The mapping is done here: `\def\btit!`⟨*a-mark*⟩`{`⟨*title*⟩`}`, `\def\f!`⟨*a-mark*⟩`{`⟨*b-mark*⟩`}`.

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

op-bible.opm

```
\outer\def\BookTitle #1 #2 #3{\sxdef{btit!#2}{#3}\sxdef{f!#2}{#1}}
```

The `\BookException` macro saves a code which is used in`\processbooks` loop in the group before files are read. You can redefine some filenames or something more special here.

op-bible.opm

```
\outer\def\BookException #1 #2{\global\sdef{bex!#1}{#2}}
```

The `\processbooks` macro does two loops over all `\printedbooks`. The `\printedbooks` list can or cannot be finalized by a space. The first loop body sets `\pbook!`⟨*a-mark*⟩ used for hyperlinks. The second loop body does:

- Defines `\bmark` as ⟨*b-mark*⟩ (a mark of the book used in file names)
- Defines `\amark` as ⟨*a-mark*⟩ (an actual mark of the book used in text)
- Defines `\btit` as the book title.
- Calls `\bex!`⟨*a-mark*⟩ in order to set something extra.
- Calls `\BibleBook{`⟨*title*⟩`}{`⟨*a-mark*⟩`}`
- Prints title of the book to the terminal and to the log.
- Inputs format definition file.
- Inputs notes file.
- Inputs txs file with original text of the Bible using `\bibleinput`, i.e. prints the text.

op-bible.opm

```
\def\processbooks {\par
   \ea\processbooksA \printedbooks\ignoreit. {}
   \ea\processbooksB \printedbooks\ignoreit. {}
}
\def\processbooksA #1 {%
   \if\relax#1\relax \else \sxdef{pbook!#1}{}\ea\processbooksA \fi
}
\def\processbooksB #1 {%
   \if\relax#1\relax \else
      \edef\amark{#1}
      \edef\bmark{\cs{f!#1}}
      \edef\btit{\cs{btit!#1}}
      \begingroup
```

```
          \ea\BibleBook\ea{\btit}{#1}
          \cs{bex!#1}
          \wterm{** \cs{btit!#1} {#1} **}
          \input{\fmtfile}
          \input{\notesfile}
          \bibleinput{\txsfile}
       \endgroup
       \ea \processbooksB
    \fi
}
```

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

op-bible.opm

```
\newcount\numvariants
\def\variants{\tmpnum=0 \afterassignment\variantsA \numvariants}
\def\variantsA{%
   \ifnum\tmpnum<\numvariants
      \advance\tmpnum by1
      \afterfi{\variantsB{\the\tmpnum}}%
   \fi
}
\def\variantsB#1#2{%
   \ifnum#1=1 \gdef\tmarkA{#2}%
   \else \sxdef{var!#1}{#2}%
   \fi
   \variantsA
}
```

`\vdef {phrase-A} {phrase-B} {phrase-C} ...` does
`\def\v!tmark-B!phrase-A{phrase-B}` `\def\v!tmark-C!phrase-A{phrase-C}` etc.

op-bible.opm

```
\def\vdef#1{\def\tmp{#1}\tmpnum=1 \vdefA}
\def\vdefA{%
   \ifnum\tmpnum<\numvariants
      \advance\tmpnum by1
      \afterfi{\vdefB{\the\tmpnum}}%
   \fi
}
\def\vdefB#1#2{\sxdef{v!\cs{var!#1}!\tmp}{#2}\vdefA}
```

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

op-bible.opm

```
\def\x/#1/{\trycs{v!\tmark!#1}{\xA#1/}}
\def\xA#1/{#1\ifx\tmarkA\undefined \else \ifx\tmark\tmarkA \else
   \printwarn{\string\x/#1/ -- this phrase is undefined by \csstring\\vdef}%
   \fi\fi
}
```

`\wdef` ⟨*chap-num:verse-num*⟩ `{phrase-A}={phrase-XA}` `{phrase-B}={phrase-XB}`
`{phrase-C}={phrase-XC}` `...` declares

```
\def\w!fv!tmark-A!phrase-A{phrase-A} \def\ww!fv!tmark-A!phrase-A{phrase-XA}
\def\w!fv!tmark-B!phrase-A{phrase-B} \def\ww!fv!tmark-B!phrase-A{phrase-XB}
\def\w!fv!tmark-C!phrase-A{phrase-C} \def\ww!fv!tmark-C!phrase-A{phrase-XC}
```

where `fv` is ⟨*full-vref*⟩. The number of parameters must be equal to `\numvariants` declared by `\variants`. The `={...}` part of parameters is optional, if it is missing then the relevant control sequence is undefined.

op-bible.opm

```
\def\wdef #1 #2{\edef\fv{\the\CommentedBook/#1}\def\phraseA{#2}\tmpnum=0 \wdefA{#2}}
\def\wdefA{%
   \ifnum\tmpnum<\numvariants
      \advance\tmpnum by1
```

```
      \ea \wdefB
   \fi
}
\def\wdefB #1{\def\tmp{#1}\isnextchar={\wdefC}{\wdefC={}}}
\def\wdefC =#1{%
   \ifcsname w!\fv!\cs{var!\the\tmpnum}!\phraseA\endcsname
      \printwarn{\noexpand\wdef used secondly for verse \vf, ignored}%
   \else  \sxdef{w!\fv!\cs{var!\the\tmpnum}!\phraseA}{\tmp}%
          \ifx^#1^\sxdef{w!\fv!\cs{var!\the\tmpnum}!\phraseA}{#1}\fi
   \fi
   \wdefA
}
```

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

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

op-bible.opm

```
\def\renum #1 #2:#3 = #4 #5..#6 {%
   \tmpnum=#3\relax
   \fornum #5..#6 \do {\sxdef{rn!#4!#2:\the\tmpnum}{##1}\incr\tmpnum}%
}
```

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

op-bible.opm

```
%% Nasledujici makra jsou zatim provizorni a nedokumentovana.

\def\chaptit#1{\ifhmode \setbox0=\lastbox \par \nobreak\vskip-\baselineskip \fi
   \medskip{\chapfont\Red#1}\endgraf\nobreak\medskip}

\def\prevtmpb{}
\def\doNote#1#2{%
   \edef\tmpb{\cs{notepre!#1}}\replstring\tmpb{-}{--}%
   \fnote{%
      \ifx\prevtmpb\tmpb \else \tmpb \enskip \global\let\prevtmpb=\tmpb \fi
      \trymakedest{n:\currverse}%
      {\bf \expandafter \ifx \csname pword!#1\endcsname \empty \else \cs{pword!#1}. \fi}%
      \cs{notetext!#1}}{\Red#2}%
}
\def\_printfnotemark{}
\def\_textindent#1{\noindent}

\newcount \chapnum
\def\source#1{}
\def\BibleBook#1#2{\def\currbook{#2}%
   \bigskip {\bookfont #1}\par\nobreak\medskip \chapnum=0 }

\def\dopsat{{\Red !!! DOPSAT !!! }}

\def\setvariant#1{}
\def\bibleinput#1 {\bgroup
   \catcode`##=13 \bgroup\lccode`~=`## \lowercase{\egroup\let~}=\processline
   \input #1
   \egroup
}
\def\_afterload{\adef<{\bref}}
\_afterload

\endinput
```