

AcroTeX.Net

**The bibpes and fetchbibpes Packages**  
**Inserting Bible Passages from e-Sword**

**D. P. Story**

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>The bibpes Package</b>	<b>4</b>
2.1	Methodologies . . . . .	5
2.2	Create the DEF file . . . . .	7
2.3	Production and organization of your TXT and DEF files . . . . .	9
2.4	e-Sword Book abbreviations . . . . .	10
<b>3</b>	<b>The fetchbibpes Package</b>	<b>11</b>
3.1	The fetch verse commands . . . . .	11
	• Fetching a verse that is not available. . . . .	13
	• The options of the fetch verse commands . . . . .	13
3.2	Package options . . . . .	25
3.3	Supporting commands for fetching verses . . . . .	27
	• Adding Bibles . . . . .	27
	• Setting the default Bible . . . . .	27
	• The <code>\useBookStyle</code> command . . . . .	28
	• <code>\usePassage</code> : The raw retrieval of a passage . . . . .	28
	• <code>\fetchversestxt</code> : raw retrieval with options . . . . .	29
	¶ The <code>typeset</code> option . . . . .	29
	¶ Cobbling together verses . . . . .	29
3.4	Formatting the translation note . . . . .	30
	• Formatting through the <code>transl</code> key . . . . .	30
	¶ Using <code>\fetchverses</code> . . . . .	30
	¶ Using <code>\fetchversestxt</code> . . . . .	31
	• Setting the default format . . . . .	31
	¶ <code>\fetchverses</code> . . . . .	31
	¶ <code>\fetchversestxt</code> . . . . .	31
3.5	Setting various spaces . . . . .	32
	• The space prior to the reference for <code>citeatend</code> . . . . .	32
	• Setting the space after the book name . . . . .	32
	• The space following the passage reference . . . . .	33
	• The space between the book and book number . . . . .	33
3.6	Formatting the verse numbers . . . . .	34
3.7	Dealing with <i>Strong's Concordance</i> numbers . . . . .	35
3.8	The CFG file of the package . . . . .	36
3.9	Marginal notes . . . . .	37
3.10	Retrieving only the verses needed . . . . .	38
3.11	The <code>\sr</code> command . . . . .	39
3.12	The <code>declareBVs</code> and <code>declareBVs*</code> environment . . . . .	39
3.13	Fetching a substring of a verse . . . . .	40
3.14	Some extended quote environment . . . . .	43
3.15	Troubles with quotation marks and apostrophes . . . . .	44

Table of Contents (cont.)	3
4 <b>Combining bibpes and fetchbibpes</b>	46
5 <b>My retirement</b>	46

## 1. Introduction

The motivation for this pair of packages (`bibpes` and `fetchbibpes`) came from my friend Thomas Price, who was preparing Bible studies using a combination of the application e-Sword (<http://www.e-sword.net/>) and  $\LaTeX$ . He wanted a ‘database’ of Bible passages from which he could simply ‘fetch’ passages into the  $\LaTeX$  source file. To that end, the pair of packages `bibpes` and `fetchbibpes` were written.

- `bibpes` is the ‘database’ part of the problem. Use the e-Sword application to copy and paste desired passages into an (empty) TXT file. Use `makebibpes.tex` to convert the TXT file to a DEF file formatted in a way that is usable by `fetchbibpes`.
- `fetchbibpes` is the ‘fetch’ portion. Using the `fetch` commands of `fetchbibpes` to reference the passages to be typeset into the  $\LaTeX$  source.

## 2. The `bibpes` Package

The `bibpes` package (`bible` passages from `e-sword`) takes a TXT file, and writes a DEF file suitable for use by the `fetchbibpes` package. The contents of the TXT file consists of Bible passages that have been pasted in from the e-Sword desktop application or the e-sword macro add-in to Microsoft Word. This package reads in that TXT file and writes a DEF file where each passage referenced is made into a text macro that expands to the passage.

This package correctly processes Bible verses in the *English language*. With the exception of German (using the German Elberfelder Bible (GEB), no other language has been tested.

**Package options** The package has several options, listed in display that follows.

```
\usepackage[%
  infolder=<infolder>,
  outfolder=<outfolder>,
  txtfilein=<base_name_txt>,
  deffileout=<base_name_def>,
  build|!build
]{bibpes}
```

### Description of options

`infolder=<infolder>` The value `<infolder>` is the path to the folder location of the TXT file to be processed. If `infolder` is not specified, the TXT file is assumed to be in the source folder of the `makebibpes.tex` file (more on this file later); otherwise, the TXT file should be found in `<infolder>`.

`outfolder=<outfolder>` The value of `<outfolder>` is the path to the folder location where the DEF file is to be placed. If `outfolder` is not specified, the DEF file is placed in the source folder of the `makebibpes.tex` file (more on this file later); otherwise, the DEF file is placed in `<outfolder>`.

`txtfilein=(base_name_txt)` is the base name of the input TXT file, do not add the extension. For example, `txtfilein=myverses`; the file `myverses.txt` must be located in the folder specified by `infolder` or in the current folder if `infolder` is not specified.

A value for `txtfilein` is required for this package to do its thing; otherwise, this package does nothing.

`deffileout=(base_name_def)` is the base name of the output DEF file, do not add the extension. If `deffileout` is not specified, then the base name of the output file is the same as the base name of the input file. The new DEF file is saved in the folder specified by the `deffileout` option, or if that option is not specified, it is saved in the current folder.

`build|!build` Two esoteric options that determine whether the DEF file is created (`build`) or not (`!build`). The default is `build` and need not appear in the option list, except in contrast to `!build`. By default (`build`), `bibpes` takes `txtfilein` and produces `deffileout` file; when `!build` is specified, the `deffileout` file is not produced. See [Section 2.2](#) for additional remarks on `build` versus `!build`.

**Steps to create a DEF file from a TXT file.** There are two steps (1) create a TXT file containing the Bible passages to be referenced; (2) run the `makebibpes.tex` file with `txtfilein` and `deffileout` set as desired. For step (1) read [Section 2.1](#) on ‘Methodologies’; for step (2), read the paragraph ‘Create the DEF file’ on page 7.

## 2.1. Methodologies

To create a TXT file consisting of Biblical verses obtained from the e-Sword application, there are two methodologies to consider.

**Methodology 1.** Work directly with the e-Sword desktop application, located at URL <http://www.e-sword.net/>. In this workflow, we copy and paste the verses directly into a TXT file.

1. Open your favorite T<sub>E</sub>X editor, create a new document or open an existing TXT file. If your T<sub>E</sub>X editor can automatically wrap long lines, *turn this feature off* before you paste any verses into the document.
2. Open the e-Sword desktop application. Select Bible > Copy Verses...
3. From the Copy Verses dialog box, refer to [Figure 1](#), choose the verses required. Select the following options from the dialog:
  - Abbreviated book names option: check this option
  - Include translation name option: this box *must be checked*.
  - Include text formatting option: clear this checkbox
  - Format option: Be sure to use the Format labeled as 1.
  - Reference delimiter: Keep the reference delimiters of parentheses ‘(’ and ‘)’. The delimiters are required to be parentheses as they are used in the definition of the Bible verses and use in the `fetchbibpes` package.

4. Press the Copy button.
5. Paste the copied verses into your TXT file. **Important!** Verify that each verse is on a single line, do not allow your editor to wrap any line around. You should have *one verse per line*.
6. When you are finished pasting verses, save the TXT file using some appropriate base name, say *myverses.txt*.

**Methodology 2.** Work within Microsoft Word and use the e-Sword ADD-IN (<http://www.e-sword.net/extras.html>). The assumptions are that you have the e-Sword application and have installed the Word add-in.

1. Open an empty Word document
2. Select ADD-INS
3. Tap the icon e-Sword Macros: Paste Verses from e-Sword.
4. See the description of step 3, in **Methodology 1** above.
5. When satisfied, save the file as a TXT file (Plain text), choose a base name that is meaningful, say *myverses.txt*
6. Place *myverses.txt* in the source folder.

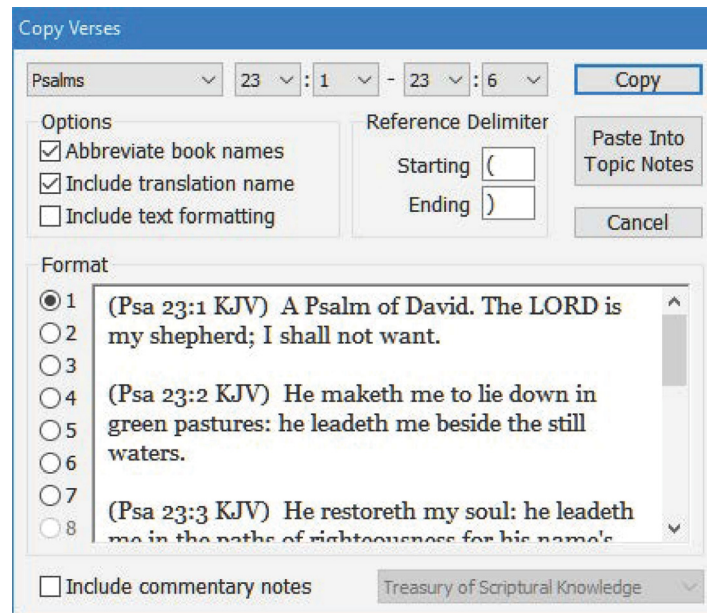


Figure 1: Copy Verses dialog box

**Other comments.** Quotations (sometimes) appear as double quotes ("), the companion package `fetchbibpes` attempts to convert double quotes to the  $\TeX$  notion of quotes; therefore, it is important to copy verses that have matched double quotes.

In some translations, Jesus' words are quoted in red and are not set off by quotation marks. You'll have to hand edit the TXT file.

```
(Mat 11:4) Jesus answered and said unto them, Go and shew John
again those things which ye do hear and see:
```

can be changed to,

```
(Mat 11:4) Jesus answered and said unto them,
\textcolor{red}{Go and shew John again those things which ye do
hear and see:}
```

to be typeset as

```
(Mat 11:4) Jesus answered and said unto them, Go and shew John again those things
which ye do hear and see:
```

That is, should such emphasis of Jesus' words be desired.<sup>1</sup>

## 2.2. Create the DEF file

After you have created your TXT file, open the file `makebibpes.tex`, which is found in the `examples` folder. This is a short file that reads:

```
1 \documentclass{article}
2 \usepackage[options]{bibpes}
3 \begin{document}
4 \reportOnBibPes
5 \readbackDefFile
6 \end{document}
(1)
```

The `makebibpes.tex` file, also found in the root folder of this distribution, is used to convert a TXT file of Bible verses to a DEF file of Bible verses. The action of this package is to take each line of Bible verse in the TXT file, for example, the verse in the TXT file,

```
(Gen 1:1 KJV) In the beginning God created the heaven and the earth.
```

and to transformed it to the following verse in the DEF file,

```
\BV(Gen 1:1 KJV) In the beginning God created the heaven and the earth.\null
```

The command `\BV` (**B**ible **V**erse) is defined in the `fetchbibpes` package. The end of the line is delimited by the `\null` command. [Section 3.12](#) discusses the general format of the argument `<verse-ref>` of `\BV`.

<sup>1</sup>Here, and throughout this manual, verses taken from the TXT or DEF files are wrapped to the next line, this is done for *display purposes only*. In both the TXT and DEF files, the rule is *one verse per line*.

It is the `fetchbibpes` package that uses this special DEF file to fetch a verse from the file and typesets it into the document. See the documentation of `fetchbibpes` in [Section 3](#) below.

The two commands `\reportOnBibPes` and `\readbackDefFile` expand to useful information when you compile the bibpes document, as seen in display (1). The first command expands to the input and output information given by the options `infolder`, `txtfilein`, `outfolder`, and `deffileout`; while `\readbackDefFile` reads back a listing of all passage just created. As a simple example,

```
\documentclass{article}
\usepackage[build,
  infolder=mytxts,
  outfolder=mydefs,
  txtfilein=Gen1-KJV
]{bibpes}
\begin{document}
\reportOnBibPes
\readbackDefFile
\end{document}
```

When compiled, the bibpes document might typeset as,

```
Input file: mytxts/Gen1-KJV.txt
Output file: mydefs/Gen1-KJV.def
```

```
Gen 1:1 KJV In the beginning God created the heaven and the earth.
Gen 1:2 KJV And the earth was without form, and void; and darkness was upon the face of the deep.
And the Spirit of God moved upon the face of the waters.
Gen 1:3 KJV And God said, Let there be light: and there was light.
Gen 1:4 KJV And God saw the light, that it was good: and God divided the light from the darkness.
Gen 1:5 KJV And God called the light Day, and the darkness he called Night. And the evening and
the morning were the first day.
```

.....

`build` Passage references appear in the margins, while the passages themselves are in the  
`verses` body of the document. When the `build` option, shown in bold font above, is specified,  
`!build` `bibpes` is fully functional in the sense that the TXT file is read and the DEF file is created  
and saved. When `build` is changed to `!build`, the typeset document displayed above  
appears, but the TXT file is *not read* and the DEF is *not created*. Using `!build` is useful  
to get a typeset listing of the passages in the DEF file, which, it is assumed, has *already  
been built*.

As an application seen later, a special command `\bibpesBody` is also defined, basically consisting of lines (3)-(6), which include the beginning and ending of the document. A revised `makebibpes.tex` document can, consequently, read as follows:

```
1 \documentclass{article}
2 \usepackage[<options>]{bibpes} (2)
3 \bibpesBody\relax
```



It is important to have the `\relax` command following `\bibpesBody`. This document should behave the same way as the other, more verbose one given previously.

[Section 4](#) contains another application to `\bibpesBody`.

### 2.3. Production and organization of your TXT and DEF files

The file `makebibpes.tex` is found in the root folder of the `fetchbibpes` distribution. Modern  $\text{\TeX}$  systems,  $\text{MiK}\text{\TeX}$  comes to mind, do not allow you to compile a source file from their application folder, and the folder itself is `readonly`. Therefore, you must move `makebibpes.tex` to another folder.<sup>2</sup>

**Production and organization.** There are two thoughts that come to mind for producing DEF files.

1. Move `makebibpes.tex` into the working folder of your project. Within the working folder, you can create your TXT and DEF files as described in the [Sections 2.1](#) and [2.2](#), respectively.

Or, for a tidier setup, you can save your TXT and DEF file in subfolders of the working folder. Create two subfolders `mytxtfiles` and `mydeffiles`. When you convert your TXT files into DEF files, specify the options `infolder=mytxtfiles` and `outfolder=mydeffiles`. For example,

```
\usepackage[%
  infolder=mytxtfiles,
  outfolder=mydeffiles,
  txtfilein=1Cor1
]{bibpes}
(3)
```

`bibpes` looks for `1Cor1.txt` in the `mytxtfiles` folder and saves the newly created `1Cor1.def` file in the `mydeffiles` folder. Later, when you use the `fetchbibpes` package, specify `deffolder=mydeffiles` on the option list of `fetchbibpes`, see [Section 3.2](#) for a description of the `deffolder` option.

2. Create a folder named `mybibleverses`, or any such name, and move the conversion file `makebibpes.tex` to this folder. This folder acts as a ‘database’ of Bible verses. As described above, create subfolders, one for TXT files and the other for DEF files (perhaps named `mytxtfiles` and `mydeffiles`).

Over time, you save your TXT files from e-Sword into the `mytxtfiles`. Convert the TXT files into DEF by compiling `makebibpes.tex` with the options in equation (3) above.

For any given project, move selected DEF files, as needed, into your the root of your working folder, or place them in a subfolder, `mydeffiles`. In the latter case, specify `deffolder=mydeffiles` on the open list of `fetchbibpes`, see [Section 3.2](#) for a description of the `deffolder` option.

<sup>2</sup>That is why another copy of `makebibpes.tex` is in the `examples` folder.

**Just a few DEFs, many, or none?** The question addressed here is whether, for a given project, develop many or a just a few DEF files for use in the project.

**Just a few DEFs.** From the e-Sword application, you can save *all the verses* you need for a given project into a *single* TXT file, then create a single DEF file. If you have a ‘database’ of DEF files already, you can copy and paste the required verses into a single DEF file.

We preview the `fetchbibpes`, in the case of a single DEF file, you can specify your only file (`lesson1`) in the option list of `fetchbibpes`:

```
\usepackage[%
  useverses=lesson1,
  deffolder=mydeffiles
]{fetchbibpes}
```

Here, `fetchbibpes` looks for `lesson1.def` in the subfolder `mydeffiles`.

This approach is useful for a small project (lesson) requiring verses of a limited scope. As you need more verses, just add them into your DEF file.

**Many DEF files.** Another choice is to use several DEF files, perhaps from your ‘database’ collection.

```
\usepackage[%
  useverses={Gen1,Mat2,Rom1,1Cor1},
  deffolder=mydeffiles
]{fetchbibpes}
```

Here, you plan to reference verses in the `Gen1.def` file (containing Genesis, chapter 1), `Mat2.def`, `Rom1.def`, and `1Cor1.def`. With this database approach, authoring Bible lessons is more modular; just drop in the DEF files you need into the appropriate folder.

The ‘database’ approach is useful for large projects. It is critical to name your DEF files to make it easy to determine which files to include in your document. In the development of this package and `fetchbibpes`, I started a naming convention: `Gen1.def` (Chapter 1, or a subset of chapter 1, of Genesis of KJV)<sup>3</sup>, `Gen1-ISV.def` (Chapter 1, or a subset of chapter 1, of Genesis of ISV)<sup>4</sup>, and so on.

**No DEF files.** When the `useverses` option does not appear, or has no value, a warning is generated and written to the log to remind you to specify a value of the `useverses` option. You can suppress this warning by specifying `useverses=none`. This declares that no DEF files are to be input. Instead, you can manage the verses by hand through the use of the `declareBVs` environment, refer to [Section 3.12](#) on page 39.

#### 2.4. e-Sword Book abbreviations

The e-Sword desktop application uses 3-letter abbreviations for the books of the Bible, see the **e-Sword Abbr** column of [Table 1](#). (39 in Old Testament and 27 in the New Testament).

<sup>3</sup>KJV is the King James Version

<sup>4</sup>ISV is the International Standard Version

Book	e-Sword		Book	e-Sword	
	Abbr	Alt_Abbr		Abbr	Alt_Abbr
Genesis	Gen	Genesis	Nahum	Nah	Nahum
Exodus	Exo	Exodus	Habakkuk	Hab	Habakkuk
Leviticus	Lev	Leviticus	Zephaniah	Zep	Zephaniah
Numbers	Num	Numbers	Haggai	Hag	Haggai
Deuteronomy	Deu	Deuteronomy	Zechariah	Zec	Zechariah
Joshua	Jos	Joshua	Malachi	Mal	Malachi
Judges	Jdg	Judges	Matthew	Mat	Matthew
Ruth	Rth	Ruth	Mark	Mar	Mark
1 Samuel	1Sa	ISamuel	Luke	Luk	Luke
2 Samuel	2Sa	IISamuel	John	Joh	John
1 Kings	1Ki	IKings	Acts	Act	Acts
2 Kings	2Ki	IIKings	Romans	Rom	Romans
1 Chronicles	1Ch	IChronicles	1 Corinthians	1Co	ICorinthians
2 Chronicles	2Ch	IIChronicles	2 Corinthians	2Co	IICorinthians
Ezra	Ezr	Ezra	Galatians	Gal	Galatians
Nehemiah	Neh	Nehemiah	Ephesians	Eph	Ephesians
Esther	Est	Esther	Philippians	Php	Philippians
Job	Job	Job	Colossians	Col	Colossians
Psalms	Psa	Psalms	1 Thessalonians	1Th	IThessalonians
Proverbs	Pro	Proverbs	2 Thessalonians	2Th	IIThessalonians
Ecclesiastes	Ecc	Ecclesiastes	1 Timothy	1Ti	ITimothy
Song of Solomon	Son	SongofSolomon	2 Timothy	2Ti	IITimothy
Isaiah	Isa	Isaiah	Titus	Tit	Titus
Jeremiah	Jer	Jeremiah	Philemon	Phm	Philemon
Lamentations	Lam	Lamentations	Hebrews	Heb	Hebrews
Ezekiel	Eze	Ezekiel	James	Jas	James
Daniel	Dan	Daniel	1 Peter	1Pe	IPeter
Hosea	Hos	Hosea	2 Peter	2Pe	IIPeter
Joel	Joe	Joel	1 John	1Jn	IJohn
Amos	Amo	Amos	2 John	2Jn	IIJohn
Obadiah	Oba	Obadiah	3 John	3Jn	IIIJohn
Jonah	Jon	Jonah	Jude	Jud	Jude
Micha	Mic	Micha	Revelation	Rev	Revelation

Table 1: e-Sword abbreviations

### 3. The fetchbibpes Package

The problem of fetching Bible verses was suggested to me by my friend Tom Price. He plans to write Bible lessons, notes and handouts prepared using  $\text{\LaTeX}$  and these packages (bibpes and fetchbibpes).

This is a companion package to bibpes. Use the commands of this package to fetch the Bible passages that the bibpes package has defined.

#### 3.1. The fetch verse commands

The major commands of this package are `\fetchverse` and `\fetchverses`, latter one being more powerful than the former. The syntax for these two are,

```

\fetchverse[⟨opts⟩]{⟨book⟩_⟨ch⟩:⟨vrs⟩} ①
\fetchverses*[⟨opts⟩]{⟨book⟩_⟨ch⟩:⟨vrs⟩} ②
\fetchverses*[⟨opts⟩]{⟨book⟩_⟨ch⟩:⟨vrs⟩-} ③
\fetchverses*[⟨opts⟩]{⟨book⟩_⟨ch⟩:⟨vrs1⟩-⟨vrs2⟩} ④

```

where *⟨opts⟩*, discussed in ‘[The options of the fetch verse commands](#)’ on page 13, are any of the supported options; *⟨book⟩* is either the three letter book abbreviation that e-Sword uses (refer to the **e-Sword Abbr** column of [Table 1](#)) or one of the alternate abbreviations, seen in the **Alt\_Abbr** column of [Table 1](#); *⟨ch⟩:⟨vrs⟩* is a chapter-verse reference. The `\fetchverse` command only fetches a *single verse*, while the `\fetchverses` (plural) command can fetch a single verse (*⟨ch⟩:⟨vrs⟩*) or can fetch *multiple verses* (as in lines ③ and ④). The run of verses must all be from the same chapter. The ‘\*’ parameter is optional, when present, the book and verse references are shown, otherwise they are not. We illustrate with `Gen 1:1`, `Gen 1:1-`, and `Gen 1:1-2`.

When the upper bound of the range is not specified, as in ③, the passages are displayed from the starting verse until the *first undefined verse* (with the same chapter number *⟨ch⟩*). This ‘open-ended run’ may go to the end of the chapter (if all verses are defined), or just run unto there are no more verses from that chapter.

We present some simple examples of the syntax variations of `\fetchverses`:

- ① `\fetchverse{Gen 1:1}`: (Gen 1:1) In the beginning God created the heaven and the earth.
- ② `\fetchverses{Gen 1:1}`: In the beginning God created the heaven and the earth.
- ③ `\fetchverses{Gen 1:1-}`: In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters. <sup>3</sup>And God said, Let there be light: and there was light. <sup>4</sup>And God saw the light, that it was good: and God divided the light from the darkness. <sup>5</sup>And God called the light Day, and the darkness he called Night. And the evening and the morning were the first day.
- ④ `\fetchverses*{Gen 1:1-2}`: (Gen 1:1-2) In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.
- ④ `\fetchverses{Gen 1:1-2}`: In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

Though the three-letter abbreviations are preferred for the book names, `fetchbibpes` does support an alternate set of names as seen in the **Alt\_Abbr** column of [Table 1](#).

- `\fetchverse{Genesis 1:1}`: (Gen 1:1) In the beginning God created the heaven and the earth.
- `\fetchverse{IKings 1:1}`: (1Ki 1:1) Now king David was old and stricken in years; and they covered him with clothes, but he gat no heat.

- `\fetchverses*{ICorinthians 1:1-2}`: (1Co 1:1-2) Paul, called to be an apostle of Jesus Christ through the will of God, and Sosthenes our brother, <sup>2</sup>Unto the church of God which is at Corinth, to them that are sanctified in Christ Jesus, called to be saints, with all that in every place call upon the name of Jesus Christ our Lord, both theirs and ours:

- **Fetching a verse that is not available.**

An error that can occur surely. For example,

```
\fetchverse{Gen 1:11}
(Gen 1:11) ??(Gen 1:11 KJV)??
```

In this case, there is no Gen 1:11, the first chapter only has 10 verses. Instead of the passage, the marker `??(Gen 1:11 KJV)??` appears. This signals that the passage is undefined.

Another possibility is that the passages are not in any of the DEF files you've input into the current document.

```
\fetchverses{Gen 2:1-3}
??(Gen 2:1 KJV)?? 2??(Gen 2:2 KJV)?? 3??(Gen 2:3 KJV)??
```

In this case, chapter 2 of the book of Genesis was not input into the current document, so `fetchbibpes` cannot find them, and marks which of the verses were not found.

Another error is simply mistyping the passage reference, for example, the declaration,

```
\fetchverse{gen 1:1} yields
(??gen?? 1:1) ??(gen 1:1 KJV)??
```

Notice that the reference label `(??gen?? 1:1)` has `'??'` around the *book name*, this means that the book name is not recognized, possibly a misspelling. In this case, 'gen' should be 'Gen'. A warning message appears in the log.

One last common mistake is to use `\fetchverse` (singular) when `\fetchverses` (plural) is needed; for example,

```
\fetchverse{ICorinthians 1:1-2} yields
(1Co 1:1-2) ??(1Co 1:1-2 KJV)??
```

In this case, we referenced a range of verses, but used `\fetchverse` to fetch. It is perhaps always better to use `\fetchverses` for single and multiple verses.

- **The options of the fetch verse commands**

In this section, we describe the `<opts>` argument of the fetch verse commands. These options control how the book names are displayed.

`abbr=<esword|none|long|short>` The `abbr` key designates how the book reference is displayed, possible values are `esword`, `none`, `long`, and `short`. Specifying `abbr` is equivalent to `abbr=esword`. Initially, we set `abbr=esword`.

### Examples

- `\fetchverse[abbr=esword]{Gen 1:1}`: (Gen 1:1) In the beginning God created the heaven and the earth.
- `\fetchverse[abbr=none]{Gen 1:1}`: (Genesis 1:1) In the beginning God created the heaven and the earth.
- `\fetchverse[abbr=long]{Gen 1:1}`: (Gen. 1:1) In the beginning God created the heaven and the earth.
- `\fetchverse[abbr=short]{Gen 1:1}`: (Gn. 1:1) In the beginning God created the heaven and the earth.

`roman=<true|false>` The `roman` key determines how book numbers are displayed, '1 Corinthians' versus 'I Corinthians', for example. Possible values are `true` (use roman letters) or `false` (use arabic numbers). Specifying `roman` without a value is equivalent to `roman=true`. Initially, we set `roman=false`.

This key has no effect when `abbr=esword`.

### Examples

- (1 Cor. 1:1) Paul, called to be an apostle of Jesus Christ through the will of God, and Sosthenes our brother,
- (I Cor. 1:1) Paul, called to be an apostle of Jesus Christ through the will of God, and Sosthenes our brother,

`from=<KJV|KJV+|ISV>` The value of the `from` key is the Bible translation to be used. For example, `from=ISV`.

**Example** `\fetchverse[abbr=none,from=ISV]{Gen 1:1}` yields,

(Genesis 1:1) In the beginning, God created the universe.

The `from` key, only supports values KJV, KJV+, ISV, these are the versions of the Bible are free versions available from e-Sword commonly used. When the `from` key is not specified, the default Bible is the `\defaultBible`, see [Section 3.3](#) for more information on setting the default Bible. Initially, the default Bible is set to KJV; thus, `\fetchverse{Gen 1:1}` fetches Genesis 1:1 from the KJV, if defined.

If you want to introduce other Bible translations into your document, use the command `\addtoBibles` to declare them; see [Section 3.3](#) for more information on `\addtoBibles`. To add Bible translations to all documents you produce, 'register' them in the configuration files; refer to [Section 3.8](#) to read how to do this.

`alt=<word>` The value `<word>` is a single word (containing no spaces). The `fetchbibpes` fetches the verses in the DEF files, as originally obtained from the e-Sword application. These verses are without formatting or emphasis or color. By hand-editing selected DEF files, you can create rich text Bible passages and retrieve them through the `alt` key. For example, we edit the file containing Gen 1:1 KJV and add a line:

```

\BV(Gen 1:1 KJV) In the beginning
  God created the heaven and the earth.\null
\BV(Gen 1:1 KJV Alt) \dots
  \emph{God created the heaven and the earth}.\null

```

(Lines are wrapped around above to fit within the margins, this *should not be done* in the DEF file itself.) To access this alternate version of Gen 1:1, we use the `alt` key.

```

\fetchverse{Gen 1:1}
(Gen 1:1) In the beginning God created the heaven and the earth.
\fetchverse[alt=Alt]{Gen 1:1}
(Gen 1:1) ...God created the heaven and the earth.

```

There can be many variations on verses, `alt=Alt`, `alt=Alt1`, `alt=Color`, and so on. When you edit a DEF file to add an alternate version, there must be a space between the Book (KJV), for example) and the *<word>* (`\BV(Gen 1:1 KJV Alt)`).

Beginning at date 2016/11/07, the behavior of the `alt` key is modified. If you specify an alternate version, say `alt=Alt` (`\fetchverse[alt=Alt]{Gen 1:1}`) and that alternate version is *not available* (undefined), `fetchbibpes` attempts to typeset the the same verse with no `alt` key, and writes an informative message to the log. The previous behavior is to mark that passage with ‘??’, which indicates the passage is undefined. This new behavior also manifests itself with the `alt*` key. The demo file `bibpes_alt-star.tex` tests this new behavior.

The command `\useOldAlt` reverts to the old behavior of `alt` and `\useNewAlt` to returns to the new behavior. The default is `\useNewAlt`.

`alt*=<list-of-words>` The parameter *<list-of-words>* is a comma-delimited list of `alt` keywords. Empty entries are supported. Perhaps an example before further explanation.

```

\fetchverses[alt*={Alt1,,Alt2,,Alt1}]{Gen 1:1-10}

```

The `fetchbibpes` package typesets the following verses: Gen 1:1 Alt1, Gen 1:2, Gen 1:3, Gen 1:4 Alt2, Gen 1:5, Gen 1:6 Alt1, Gen 1:7, Gen 1:8, Gen 1:9, Gen 1:10. If any of the `alt` versions are undefined, `fetchbibpes` typesets the passage with no `alt` key specified and an informative message is written to the log.

The demo file `bibpes_alt-star.tex` tests this feature.

`transl=<text>` The `transl` key is an annotation key that appends text (*<text>*) to the end of the Bible reference, such as `transl=KJV`, to obtain the *translation note*. The `transl` internally defines a macro that holds its value, but also defines a public command `\gtransl` that expands to *<text>*. This command is redefined at the next use of the `transl` key within `\fetchverses` or `\fetchversestxt`.

`\gtransl` defined

**Examples** We set the translation note to ‘King James Version’.

```
\fetchverse[abbr=none,transl=King James Version]{Gen 1:1}
```

(Genesis 1:1 King James Version) In the beginning God created the heaven and the earth.

Normally, the version abbreviation is used as the *<text>*:

```
\fetchverse[abbr=none,transl=KJV]{Gen 1:1}
```

(Genesis 1:1 KJV) In the beginning God created the heaven and the earth.

The translation note does not appear unless the `transl` key is specified with a value. In the example below, there is no `transl` key (and there is no `from` key so the default Bible is used (KJV)).

```
\fetchverse[abbr=none]{Gen 1:1}
```

(Genesis 1:1) In the beginning God created the heaven and the earth.

To get the translation note to appear you can either specify a value for the `transl` key, as we did above, or turn on the feature of always showing the translation note.

```
\showTranslAlways \showTranslDecld
```

`\showTranslAlways` turns on the translation note,<sup>5</sup> while `\showTranslDecld` to the default behavior, that is, show the translation note only when the `transl` key has a declared value. Thus,

```
\showTranslAlways
\fetchverse[abbr=none]{Gen 1:1} expands to,
```

(Genesis 1:1 KJV) In the beginning God created the heaven and the earth.

The Bible abbreviation is used (KJV), if the `transl` is present in the optional parameter list, its value will be used instead.

```
\showTranslAlways
\fetchverse[abbr=none,transl=KJ]{Gen 1:1} expands to,
```

(Genesis 1:1 KJ) In the beginning God created the heaven and the earth.

The sample file `fetchversestxt.tex` illustrates `\showTranslAlways`.

`from*=<version>` The `from*` combines the actions of the options `from` and `transl` and is equivalent to `from=<version>,transl=<version>`. Here *<version>* must be a recognized version of the Bible available on your computer.

**Example** `\fetchverse[abbr=none,from*=ISV]{Gen 1:1}` yields,

(Genesis 1:1 ISV) In the beginning, God created the universe.

<sup>5</sup>If the `transl` key is not specified, it uses the value of the `from` key; if the `from` key is not specified, the default Bible is used.



The value of the `from*` key must be a valid value of the `from` key.

`pcolor=<color>` Use the `pcolor` key to color the passage text. The value `<color>` is any color specification accepted by the `\color` macro of the `color` or `xcolor` package.

**Example** `\fetchverses*[pcolor=blue]{Gen 1:1-2}`

(Gen 1:1-2) *In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.*

The coloring includes the two tokens placed by the `enclosewith` key.

Note that the verse numbers are also colored; to typeset the verse number in black, say, redefine `\verseFmt`. For example

```
\renewcommand\verseFmt[1]{\normalcolor\verseFmtDef{#1}}
```

the above passage becomes,

(Gen 1:1-2) *In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.*

The general treatment of `\verseFmt` is taken up in [Section 3.6](#) on page 34.

In the option list of a `fetch` command, declaring `pcolor` (without a value) causes the passage text to be colored the default color (usually black). All changes in the option list are local to the passage, but you can use `\useBookStyle` to set 'global' options. Continue reading on this subject, beginning with the named paragraph [The `\useBookStyle` command](#) on page 28.

`pformat=<various>` Use the `pformat` key to change the formatting (or style) of the passage text. The `<various>` argument should be style declarations, such as `\bfseries`, `\itshape`, `\slshape`, `\sffamily`, and so on, but can be anything that does not take an argument.

**Example** `\fetchverses*[pformat=\slshape]{Gen 1:1-2}`

(Gen 1:1-2) *In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.*

More than one formatting command may be used; for example, to obtain bold slant passage text, we declare `pformat=\bfseries\slshape`.

See the `format` key for formatting the passage reference.

`delims=<tok1tok2>` The `delims` key takes a value of two characters (or tokens or groups enclosed in braces `{}`) used to set off the Bible reference. The default is `delimin=()`.

If `delims` appears in the option list without a value, then there are no characters delimiting the Bible passage.

### Examples

1. `\fetchverse[abbr=none,delims]{Gen 1:1}`  
Genesis 1:1 In the beginning God created the heaven and the earth.
2. `\fetchverse[abbr=none,delims={[]}] {Gen 1:1}`  
[Genesis 1:1] In the beginning God created the heaven and the earth.
3. `\fetchverse[abbr=none,delims={According to }{,}] {Gen 1:1}`:  
According to Genesis 1:1, In the beginning God created the heaven and the earth.

As seen in item 2, delimiters of ‘[]’ must be enclosed in braces because this all takes place within an optional argument, without the braces, the parsing of the optional argument fails. The `delims` expects two tokens, they need not be characters, they can be anything, as long as they are enclosed in matching braces. In item 3, we set `delims={According to }{,}`, the first token is placed before the passage reference, the second after; this results in a phrase that leads up to the passage itself.

- When `delim` is used with `\fetchverses` the star-option must be used, otherwise, `delim` has no effect.

`enclosewith=(tok1tok2|\quote)` The `enclosewith` key takes either a two-token argument or the command `\quote`. The latter is used to enclose the passage in quotation marks. The tokens are placed as follows: `<tok1>passage<tok2>`. When `\quote` is specified `<tok1>={"}` (left quote) and `<tok2>={"}` (right quote).

**Example** Enclose the passage in quotation marks:

```
\fetchverse[delims={According to }{,},enclosewith=\quote]{Gen 1:1}
According to Genesis 1:1, "In the beginning God created the heaven and the
earth."
```

```
\fetchverses*[delims={According to }{,},
enclosewith={\guillemotleft\space}{ \guillemotright}]{Gen 1:1-2}
According to Genesis 1:1-2, « In the beginning God created the heaven and the
earth. 2And the earth was without form, and void; and darkness was upon the
face of the deep. And the Spirit of God moved upon the face of the waters. »
```

The `enclosewith` key must be used wisely as some passages, depending on the translation, already contain single and/or double quotation marks.

`format=(value)` The `format` key is a general key for making formatting declarations, for example, we format the reference using `format=\bfseries\sffamily`.

**Example** `\fetchverse[format=\bfseries\sffamily]{Gen 1:1}` produces the passage: **(Gen 1:1)** In the beginning God created the heaven and the earth.

The formatting only effects the passage label (reference) **Gen 1:1** and its surrounding delimiters.

`caps=(true|false)` Specifying the `caps` key displays the chapter name (the passage reference or label) in capital letters ‘GENESIS’ versus ‘Genesis’. Specifying `caps` without a value is equivalent to `caps=true`.

**Example** Capitalize the full book name and while we’re at it we format the passage reference as well.

```
\fetchverse[format=\bfseries\sffamily,abbr=none,caps]{Gen 1:1}
(GENESIS 1:1) In the beginning God created the heaven and the earth.
```

`showfirst=(true|false)` The `showfirst` key is a Boolean, which if `true`, the first verse number is typeset. The default is `false`, the first verse number is not typeset. Specifying `showfirst` without a value is equivalent to `showfirst=true`.

**Examples** We compare the passages Gen 1:1-2 with and without first verse number:

- `\fetchverses[showfirst]{Gen 1:1-2}`  
<sup>1</sup>In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.
- `\fetchverses{Gen 1:1-2}`  
 In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

`shownone` When this valueless key is specified, no verse numbers are shown for the passage. In this case, the `showfirst` key has no effect.

**Example** `\fetchverses[shownone]{Gen 1:1-3}` fetches the passage, no verse numbers are shown.

In the beginning God created the heaven and the earth. And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters. And God said, Let there be light: and there was light.

`useperiod=(true|false)` The `useperiod` key is a Boolean, which if `true` a period follows any abbreviation of a book name. If `useperiod=false` no period is affixed. The default is `true`. (‘Matt.’ versus ‘Matt’, for example.)

**Examples**

- `\fetchverse[abbr=long]{Mat 2:1}`  
 (Matt. 2:1) Now when Jesus was born in Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to Jerusalem,
- `\fetchverse[abbr=long,useperiod=false]{Mat 2:1}`  
 (Matt 2:1) Now when Jesus was born in Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to Jerusalem,

The period is used by default, you must say `useperiod=false` to remove it. This option is not obeyed when `abbr=esword` (or just `abbr`, the default).

`citeatend=(true|false)` The `citeatend` key is a Boolean, which if `true`, places the citation reference at the end of the passage. Specifying `citeatend` without a value is equivalent to `citeatend=true`.

**Example** `\fetchverses*[citeatend,showfirst]{Gen 1:1-2}`

<sup>1</sup>In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters. (Gen 1:1-2)

It is possible to control the positioning of the citation using additional features, yet to appear in the record, refer to [Section 3.3](#).

`nocite` The `nocite` key is only obeyed with the `\fetchverse` command. If present, the citation for the passage does not appear.

**Example** `\fetchverse[nocite]{Gen 1:1}` displays as:

In the beginning God created the heaven and the earth.

Of course, the citation would normally appear prior to the appearance of the passage. The above passage is equivalent to `\fetchverses{Gen 1:1}`.

For `\fetchverse`, the citation appears by default, you must force it away with `nocite`; for `\fetchverses`, the citation *does not appear by default*, to make it appear, use `\fetchverses*`.

`priorref=(value)` The value of `priorref` is placed just before the book reference (`(value){\book@passage}`). The book reference itself (`\book@passage`) is enclosed in braces so `(value)` can be a macro with one argument.

**Example** Place the passage reference in the margin. The command `\mpfmt` is defined locally, within the quote environment.

Gen 1:1-2 `\fetchverses*[priorref=\mpfmt,delims]{Gen 1:1-2}`  
 In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

The definition of `\mpfmt` is made prior to the passage (and is not a command defined by the `fetchbibpes` package).

```
\def\mpfmt#1{\afterRef{}}\marginpar{\small\raggedleft#1}}
```

The `\afterRef` command is described later in this documentation.

`\ltxquotes` If this key is present (it has no value), it is assumed that the corresponding passage or passages uses  $\LaTeX$  style quotes, as in ``God's grace'` or ``God's grace'`. Such a situation can occur if you've hand-edited the DEF file that contains the passages of interest. In this case you want to use the `\ltxquotes` key with the `\fetchverses` command; for example,

```
\fetchverses*[\ltxquotes,from*=NKJV]{Mat 8:5-13}
```

(Mat 8:5-13 NKJV) Now when Jesus had entered Capernaum, a centurion came to Him, pleading with Him, <sup>6</sup>saying, "Lord, my servant is lying at home paralyzed, dreadfully tormented." <sup>7</sup>And Jesus said to him, "I will come and heal him." <sup>8</sup>The centurion answered and said, "Lord, I am not worthy that You should come under my roof. But only speak a word, and my servant will be healed. <sup>9</sup>For I also am a man under authority, having soldiers under me. And I say to this one, 'Go,' and he goes; and to another, 'Come,' and he comes; and to my servant, 'Do this,' and he does it." <sup>10</sup>When Jesus heard it, He marveled, and said to those who followed, "Assuredly, I say to you, I have not found such great faith, not even in Israel! <sup>11</sup>And I say to you that many will come from east and west, and sit down with Abraham, Isaac, and Jacob in the kingdom of heaven. <sup>12</sup>But the sons of the kingdom will be cast out into outer darkness. There will be weeping and gnashing of teeth." <sup>13</sup>Then Jesus said to the centurion, "Go your way; and as you have believed, so let it be done for you." And his servant was healed that same hour.

This particular passage has been hand-edited, and uses  $\LaTeX$  quotation marks.

There are two commands related to `\ltxquotes`, these are `\LaTeXQuotesOn` and `\LaTeXQuotesOff`, the default is `\LaTeXQuotesOff`. The option `\ltxquotes` executes `\LaTeXQuotesOn` within a group. In extraordinary situations, these commands may be explicitly expanded by the document author.

Refer to Section 3.15 for a discussion of quotation marks and apostrophes.

```
replace={\find1}\{replc1}\{find2}\{replc2}\dots\{findk}\{replck}
```

The key performs (multiple) find and replace operations. There must be an even number of arguments. For each pair, `\findi` is searched for and replaced by `\replci`. This is an exact match search. A command in the `\replci` argument might need to be protected using `\protect`.

For example,

```
\fetchverse{Mat 2:1}
```

(Mat 2:1) Now when Jesus was born in Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to Jerusalem,

As a stand alone quote, you'll note that the passage ends with a comma; we want to change this to a period. Also, just to demonstrate multiple find and replace, we'll also replace every instance of 'Jesus' with 'Jesus'.

```
\fetchverse[replace={Jerusalem,}{Jerusalem.}
{Jesus}{\textcolor{red}{Jesus}}]{Mat 2:1}
```

(Mat 2:1) Now when **Jesus** was born in Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to Jerusalem.

There are two commas in the sentence, so we must characterize the comma we want to replace, the one that follows ‘Jerusalem’.

**Rule:** Do *plain text replacements first* followed by any formatted text replacements last.

```
\fetchverse[replace={Jerusalem,}{Jerusalem.}
{Jerusalem}{\textbf{Jerusalem}}]{Mat 2:1}
```

(Mat 2:1) Now when Jesus was born in Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to **Jerusalem**.

succeeds whereas,

```
\fetchverse[replace={Jerusalem}{\textbf{Jerusalem}}
{Jerusalem,}{Jerusalem.}]{Mat 2:1}
```

(Mat 2:1) Now when Jesus was born in Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to **Jerusalem**,

fails. Notice the comma (,) was not replaced by a period (.). After the first replacement, where is no string ‘Jerusalem,’ there is ‘\textbf{Jerusalem},’.

Additional examples are found in the sample files `bibpes_replace.tex` and in `bibpes_replaces.tex` that accompany this distribution. Examples include some ‘creative’ use of `replace`.

**The `replace` key and `\fetchverses`.** A more detailed understanding of how `replace` works when retrieving multiple verses with `\fetchverses` is needed. The `replace` algorithm is run on *each verse* individually – not on the combined verses – as it is retrieved. This fact leads us to another rule.

**Rule:** The *find* text should only match text within a single verse, not across several verses, for such a match will never be successful. Restrict your searches to match fragments within a single verse.

There is a command that may be useful with the `replace` key.

```
\gobbleto{<word>}{<\cmd>}}
```

The command defines another command (`\cmd`), the action of this command is to gobble up all token between it and the *word*. The need for such a command arose when *find* text contains one or more of the active characters. (The active characters are the single left quote (‘), the single right quote or apostrophe (’), and the double quote (”).) It is difficult to make things work (find/replace) when there is one of these active characters in the *find* text. Below is an example from `bibpes_replaces.tex`. We quote a passage from Gal 1:15-16:

```
\fetchverses*[from*=NKJV]{Gal 1:15-16}
```

(Gal 1:15-16 NKJV) But when it pleased God, who separated me from my mother's womb and called me through His grace,<sup>16</sup> to reveal His Son in me, that I might preach Him among the Gentiles, I did not immediately confer with flesh and blood,

We want to annotate this passage by removing unwanted sentence fragments, so we can concentrate on the teaching of the passage. Note that mother's womb is part of verse 15, the problem is to deal with the apostrophe.

```
1 \gobbleto{and}{\gobbletoand}\gobbleto{blood,}{\gobbletoblood}
2 \fetchverses*[from=NKJV,replace=
3   {But when}{\leavevmode\ldots when}
4   {who separated me from my}{who\gobbletoand}
5   {, that I might preach Him}{\ldots\gobbletoblood}
6 ]{Gal 1:15-16}
```

Prior to the passage, we make two declarations `\gobbleto{and}{\gobbletoand}` and `\gobbleto{blood,}{\gobbletoblood}`, and insert them into our `<replace>` text, as seen above.

(Gal 1:15-16) ...when it pleased God, who called me through His grace,<sup>16</sup> to reveal His Son in me ...

In line (4) of the verbatim listing above, we find the text `who separated me from my` (could have been shorter) and replace it by `who\gobbletoand`. The `\gobbletoand` command is inserted just after 'who', and will gobble everything that follows it up to and including the word 'and'. Included in the gobbled text is the active character ', which now causes no grief. In line (5), another 'gobble-replacement' is performed for a very long passage (containing no active characters). In this way, Biblical fragments may be displayed.

Obviously, great care must be taken to prevent unwanted replacements. The `<find>` text must characterize what is to be found. Always check your work to make sure you are getting the targeted replacement and not spurious replacement.

The `replace` option is a quick and easy way of making minor editorial adjustments to the passage. Major revisions, such as putting the Jesus' words in red as some translations do, require editing the DEF file and creating alternate passages; for more information on how to create alternate passages, refer to the `alt` key on [page 14](#).

`localdefs=<localdefs>` Use the `localdefs` key to pass macro definitions into the group in which `\fetchverse` and `\fetchverses` operate. Any definitions, or other  $\LaTeX$  constructs, should not take up any space, nor should they disturb the  $\LaTeX$  code of `\fetchverse` and `\fetchverses`. The application of this key is to the `\gobbleto` command. Pass any `\gobbleto` declarations through the `localdefs` key to make them local to the group.

The 'Gal 1:15-16' passages above were made within a quote environment, which made the `\gobbleto` declarations local; to quote the same passage outside of an enclosing group, yet avoiding a global declaration, use the `localdefs` key.

```

\fetchverses*[from=NKJV,
  localdefs={\gobbleto{and}{\gobbletoand}%
    \gobbleto{blood,}{\gobbletoblood}},
  replace=
    {But when}{\leavevmode\ldots when}
    {who separated me from my}{who\gobbletoand}
    {, that I might preach Him}{ \ldots\gobbletoblood}
]{Gal 1:15-16}

```

This code expands to: (Gal 1:15-16) ...when it pleased God, who called me through His grace,<sup>16</sup> to reveal His Son in me ...

Note that because `\gobbleto{blood,}` contains a comma, we must enclose the value of `localdefs` in braces so `xkeyval` will parse it correctly.

The above example appears in `bibpes_replaces.tex`.

`typeset=<opts>` This option was designed for `\fetchversestxt`, but may be used with `\fetchverses` as well.<sup>6</sup> Refer to the subsection titled ‘`\fetchversestxt: raw retrieval with options`’ on page 29 for more information on the command.

- **For `\fetchverses`:** When `typeset` appears within the optional argument of `\fetchverses` its value (`<opts>`) is added *to the end of the option list*. This may cause options specified by `typeset` to overwrite the options that are already declared. However, there is an important exception, that for the `replace` key; in this case, the arguments of the `replace` declared within `typeset` are appended to the arguments of the `replace` declared in the option list of `\fetchverses`. Thus,

```

\fetchverses[typeset={{brother}{\textbf{brother}}},<other-opts>},
  replace={brother,}{brother.},...]{1Co 1:1}

```

This is equivalent to

```

\fetchverses[replace={brother,}{brother.}
  {brother}{\textbf{brother}},...,<other-opts>]{1Co 1:1}

```

- **For `\fetchversestxt`:** If the `typeset` key appears (possibly with an empty value) within the option list of `\fetchversestxt`, it declares that the two macros `\versestxt` and `\passagetxt` (created by `\fetchversestxt`) are intended to be `typeset` into the document; the `<opts>` value *is ignored*. The key itself effects the treatment of quotation marks and apostrophes; when present, ‘smart quotes’ are used and `\bDQ`, `\eDQ`, `\bSQ`, and `\eSQ` have effect. See [Section 3.15](#) for a discussion of these latter commands.

The purpose of `typeset` (as well as `typeset*`) is to enable the document author to write one set of options for both `\fetchverses` and `\fetchversestxt`. The value of `typeset` provides additional formatting options not available through `\fetchversestxt`.

<sup>6</sup>This key is not used with `\fetchverse`.

replace option  
special case



`typeset*=(opts)` This key has the same features as `typeset` when specified within the option list of `\fetchverses`. Within the option list of `\fetchverses.txt`, however, this key-value has no effect at all; `\fetchverses.txt` has its normal characteristics: no  $\LaTeX$  formatting or font changes are supported; the ‘smart quotes’ are not used; and the commands `\bDQ`, `\eDQ`, `\bSQ`, and `\eSQ` are not obeyed.

The section ‘[\fetchverses.txt: raw retrieval with options](#)’ on page 29 includes some discussion and examples of these last two keys.

**Predefined options using a ‘helper’ command.** It can become tedious to type the various options, nothing can be done about that, but if you use the same collection of options consistently, you can package them in a ‘helper’ command:

```
\newcommand{\Opsi}{pcolor=blue,format=\bfseries,abbr=none}
```

Then you can say,

```
\fetchverses*[\Opsi,<more_options>]{Gen 1:1-2}
```

**Important.** The placement of the ‘helper’ command is important, it should be placed just to the right of the left bracket ([), as shown above. The first token (`\Opsi`) is expanded using the  $\TeX$  primitive `\expandafter`, which is a limited kind of expansion.

```
\fetchverses*[\Opsi,caps]{Gen 1:1-2}
```

**(GENESIS 1:1-2)** In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

### 3.2. Package options

The basic syntax for `fetchbibpes` is as follows:

```
\usepackage[%
  useverses={\list},
  deffolder=<path>,
  useselverses,
  usesr,
  ignorecfg,
  cfg=<name>
]{fetchbibpes}
```

#### Description of options

`useverses={\list}` The `useverses` option is the way the DEF files are input into the document. The value of `useverses` is `{\list}`, a comma-delimited base names for DEF file created by `bibpes`. The listed base name DEF files contain verses that are to be referenced and fetched from the files. These files should be located in the folder containing the source file of the document or in the search path of your  $\LaTeX$  distribution. For example,

```
\usepackage[useverses={Gen1,Mat2,Rom1,1Cor1}]{fetchbibpes}
```

The items listed in the *comma-delimited* list are the base names of the DEF files to be input. The above assumes the listed files (Gen1.def, Mat2.def, Rom1.def, 1Cor1.def) are in the folder of the source file, or in the search path of your  $\LaTeX$  system.

`useverses=none` There is a second form for the `useverses` option where the value is not a *list*, but the word 'none'. When `useverses` does not appear in the option list, or has no value, a  $\LaTeX$  warning is generated to remind you to specify one or more DEF files. To suppress this warning, use the special of 'none', in which case no DEF files are expected to be input into the document. When using this option, you normally manually manage the Biblical passages by using the `declareBVs` environment, which is covered in [Section 3.12](#) on page 39.

`deffolder=<path>` Through `deffolder` key you can specify the *<path>* to the DEF files. Ideally, *<path>* is a subfolder of your source file. Do not place a forward slash ('/') at the end of the path, this is done internally only if *<path>* is nonempty.

```
\usepackage[%
  useverses={Gen1,Mat2,Rom1,1Cor1},
  deffolder=mydeffiles
]{fetchbibpes}
```

Here,  $\LaTeX$  looks for the listed files (Gen1.def, Mat2.def, Rom1.def, 1Cor1.def) in the folder `mydeffiles`, which is a subfolder of the source file.

This key is useful if you want to remove the clutter of numerous DEF files from the root folder of your source file.

`useselverses` is an option that tries to economize memory usage, refer to [Section 3.10](#) for details.

`fetchsubtr` is an option that includes code to extract substrings from a verse. See [Section 3.13](#) for details.

`usersr` is a specialize option that brings in the definition of the `\sr` command, see [Section 3.11](#) for more information.

`ignorecfg` When this key is present (no value is required), the CFG file for this package (`fbpes.cfg`) is *not loaded*. Refer to [Section 3.8](#) for more information on the CFG file.

`cfg=<name>` The value of this key specifies the base name of the configuration file (*<name>.cfg*) to be used. If the `cfg` does not appear or has no value, the default configuration file is used (`fbpes.cfg`). The configuration file is input only if the `ignorecfg` option *does not appear* on the package option list. An example of the default configuration file (`fbpes.cfg`) is provided with this distribution.

**Demo file:** `bibpes_tst.tex`.

### 3.3. Supporting commands for fetching verses

- **Adding Bibles**

Suppose you want to fetch verses from the American Standard Version (ASV) and Young's Literal Translation (YLT), first download them (if not already on your system) using the e-Sword desktop application; second, use the `\addtoBibles` command to add them to the list of supported Bible versions.

```
\addtoBibles{<list>}
```

This command is placed in the preamble, `<list>` is a comma delimited list of e-Sword Bible translation abbreviations. (These are apparent when you make your choice of Bible version to download.) For example, place

```
\addtoBibles{ASV,YLT}
```

in the preamble to specify the American Standard Version (ASV) and Young's Literal Translation (YLT), assuming these versions are available on your system.

**Example** `\fetchverses[from=YLT]{Gen 1:1-2}` yields,

In the beginning of God's preparing the heavens and the earth-<sup>2</sup>the earth hath existed waste and void, and darkness is on the face of the deep, and the Spirit of God fluttering on the face of the waters,

- **Setting the default Bible**

You can set your own default Bible using the `\defaultBible` command.

```
\defaultBible{<trans1>}
```

where `<trans1>` is the name of a Bible translation assigned by the e-Sword application. The name `<trans1>` must be a recognized Bible translation: KJV, KJV+, ISV, or any Bible translation registered by the `\addtoBibles` command. The command `\fbdefaultBible` holds the (current) default Bible abbreviation.

Other than the ones initially defined (KJV, KJV+, and ISV), a Bible translation can be added on with the `\addtoBibles` command, or by adding in a new Bible translation in the CFG; see [Section 3.8](#) for more information on how to do this.

**Example** Purchase, download and install NKJV (a premium Bible) and declare in a document:

```
\addtoBibles{NKJV}
\defaultBible{NKJV}
```

With these declarations, the command `\fetchverse{Gen 1:1}` fetch Gen 1:1 from NKJV, assuming such verses exist in any of the DEF files that are specified by the `useverses` option.

`\fbdefaultBible`  
defined

- **The `\useBookStyle` command**

You can design your own default key-values using the `\useBookStyle` command.

```
\useBookStyle{<opts>}
```

where `<opts>` is a comma-delimited list of key-values for the fetch verses commands. This command may be used in the body of the document to change the style.

**Example** Declaring `\useBookStyle{abbr=none,roman}\fetchverse{1Co 1:1}` produces the following passage:

(I Cor. 1:1) Paul, called to be an apostle of Jesus Christ through the will of God, and Sosthenes our brother,

The fetchbibpes package uses `\useBookStyle` to initially declare,

```
\useBookStyle{abbr,roman=false,from,transl}
```

The changes are local to any group in which they are made.

As with the fetch command options, see **Predefined options using a ‘helper’ command** on page 25, the first token in the argument of `\useBookStyle` is expanded using `\expandafter`, this means that the first token can be a macro command of key-values; for example, `\useBookStyle{\Opsi,caps}` sets the book style to your favorite options (`\Opsi`), plus the caps option.

When declared outside a group, the changes are considered global, which may not always be what is wanted. For example, if we declare `\useBookStyle{pcolor=blue}`, then all passages are colored blue by default.

```
\useBookStyle{pcolor=blue}
```

`\fetchverse{Gen 1:1}` results in (Gen 1:1) **In the beginning God created the heaven and the earth**. This may not be what is wanted within a paragraph presentation. To revert to default black passage text, use `pcolor` with no value. Thus, `\fetchverse[pcolor]{Gen 1:1}` yields (Gen 1:1) In the beginning God created the heaven and the earth.

- **`\usePassage`: The raw retrieval of a passage**

The `\fetchverse` and `\fetchverses` command are rather complex in that they must process any of the many options before it produces the passage. There may be an application to simply retrieving the passage without any frills; in this case use `\usePassage`:

```
\usePassage{<verse-ref>}
```

where `<verse-ref>` is the argument of the `\BV` command as it is specified in one of the DEF files. Thus, if we type `\usePassage{Gen 1:1 KJV}`, we get,

In the beginning God created the heaven and the earth.

That is, you get the ‘raw’ passage without any annotations (translation, book, chapter, verse, and so on). Only one passage (verse) may be retrieved this way.

- **\fetchversestxt: raw retrieval with options**

One user, a former friend, requested something more sophisticated than `\usePassage`. He wanted a `\fetchverses`-type command, but with no typesetting commands embedded, just pure text. He would then pass the text as the argument of another command.

```
\fetchversestxt[<opts>]{<book>_<ch>:<vrs>}
\fetchversestxt[<opts>]{<book>_<ch>:<vrs>-}
\fetchversestxt[<opts>]{<book>_<ch>:<vrs12

```

Each of these defines two macros: `\versetxt`, `\passagetxt`

The `\fetchversestxt` command does not expand to typeset content, rather, it defines two commands `\versetxt`, which contains the verse reference, and `\passagetxt`, which contains the passage for that verse. For example, `\fetchverses{Gen 1:1-2}` expands to nothing typeset; however, the command `'\versetxt'` expands to 'Gen 1:1-2' and the command `'\passagetxt'` expands to 'In the beginning God created the heaven and the earth. 2) And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.' The next expansion of `\fetchversestxt` overwrites the `\versetxt` and `\passagetxt` commands. You can use `\fetchversestxt` with the usual options, any formatting options are ignored (I hope), otherwise, all options should work as described earlier.



Gen 1:1-

The application my former friend had in mind was to pass the `\passagetxt` command to an annotation macro such as `\annotpro` (annot\_pro package) or `\pdfcomment` (pdfcomment package). What you see in the margin is `\annotpro`.

```
The application\fetchversestxt[showfirst]{Gen 1:1-}\annotpro[type=text,
name=Note,color=webyellow,margin,readonly,
title={\versetxt},marginwidth={\hfill\versetxt},
margprior={\hfill}]{\passagetxt}
my former friend had in mind ...
```

Refer to the manual, `annotpro_man.pdf` for the particulars of using `\annotpro` from the `annot_pro` package.

¶ **The typeset option.** The `\fetchversestxt` command accepts `typeset` (see page 24 for a description) as an option. Without the `typeset` option, `\passagetxt` does not contain any  $\text{\LaTeX}$  styling or font changes; if the `typeset` option is taken, `\passagetxt` expands to the same passage as its non-tx counterpart. The accompanying demonstration file `fetchversestxt.tex` extensively discusses the `typeset` option.

The demonstration file for this feature is `fetchversestxt.tex`. It includes several examples of the use of `typeset` and `typeset*` with both commands `\fetchverses` and `\fetchversestxt`, with and without a value for the key.

¶ **Cobbling together verses.** In this paragraph the `\cobblevrs` command is discussed and illustrated.

```
\cobblevrs*{\<cmd>}{<text>}
```

`\cobblevrs` takes  $\langle text \rangle$  and adds it to the end of the text macro  $\langle \backslash cmd \rangle$ . If  $\langle \backslash cmd \rangle$  is undefined initially, `\cobblevrs` defines it as to be empty; if  $\langle \backslash cmd \rangle$  is already defined, `\cobblevrs` adds  $\langle text \rangle$  to  $\langle \backslash cmd \rangle$ . If  $\langle \backslash cmd \rangle$  is already defined (perhaps used previously by `\cobblevrs`), to empty its contents before using again (with `\cobblevrs`) use the star-version of the command.

The `\cobblevrs` command was designed to be used with the commands `\versetxt` and `\passagetxt` generated by `\fetchversetxt`, but can be use for any purpose. Example of intended usage:

```
\fetchversetxt[typeset,abbr=none]{Gen 1:1}\cobblevrs*\holdverses
  {\versetxt\space\passagetxt}
\fetchversetxt[typeset,from*=ISV]{Gen 1:1}\cobblevrs\holdverses
  { Now, the same quote from ISV: \versetxt\space\passagetxt}
\fetchversetxt[typeset,from*=YLT]{Gen 1:1}\cobblevrs\holdverses
  { Again, but now from YLT: \versetxt\space\passagetxt}%
\holdverses % expand our cobbled verses
```

The `\holdverses` macro then expands to the following paragraph:

(Genesis 1:1) In the beginning God created the heaven and the earth. Now, the same quote from ISV: (Gen 1:1 ISV) In the beginning, God created the universe. Again, but now from YLT: (Gen 1:1 YLT) In the beginning of God’s preparing the heavens and the earth-

`\cobblevrs` is a tool that is available if ever needed. The demo file for this feature is `fetchversetxt_cobble.tex`.

### 3.4. Formatting the translation note

By “translation note”, we mean the value of the `transl` key. By default, the note appears as plain text following the Bible reference and a space.

```
\fetchverses*[from=ISV,transl=ISV]{Gen 1:1}
(Gen 1:1 ISV) In the beginning, God created the universe.
```

- **Formatting through the `transl` key**

In the next two paragraphs, we comment on formatting the translation note using the `transl` key.

¶ **Using `\fetchverses`.** The translation note may be formatted for *individual passages* using the `transl` key. *Simple textual formatting* is possible, such as

```
\fetchverses*[from=ISV,transl={- ISV}]{Gen 1:1}
(Gen 1:1 - ISV) In the beginning, God created the universe.
```

You can change font styles as well.

```
\fetchverses*[from=ISV,transl=\textbf{ISV}]{Gen 1:1}
(Gen 1:1 ISV) In the beginning, God created the universe.
```

¶ **Using `\fetchversestxt`.** The command `\fetchversestxt` does not produce any typeset material, but defines to macros `\versetxt` and `\passagetxt`. Simple changes to the translation note are no problem:

```
\fetchversestxt[from=ISV,transl={- ISV}]{Gen 1:1}
\versetxt now expands to Gen 1:1 - ISV
```

To individually change the font styling, we have to protect style change with `\protect`,

```
\fetchversestxt[from=ISV,
  transl={- \protect\textbf{ISV}}]{Gen 1:1}
\versetxt expands to Gen 1:1 - ISV
```

Above, the `\protect` command is shown in bold font for emphasis.

#### • Setting the default format

Another way to set the formatting of the translation note is by changing the default formatting. (The default is plain text preceded by a space.)

¶ **`\fetchverses`** As mentioned above, the default format is just plain text, though individual changes can be made through the `transl` key. To change the default formatting for `\fetchverses` and `\fetchverse`, use the `\translFmt` command.

```
\translFmt{\<fmt-transl-#1>}
\translFmtDef (\translFmt{\space#1})
```

Within the argument of `\translFmt`, use `#1` to symbolically represent the value of the `transl` key. The default declaration for `\translFmt` is given to the right in parentheses. For example,

```
\translFmt{ - \textbf{#1}} (Note space after left brace)
\fetchverses*[from*=ISV]{Gen 1:1}
(Gen 1:1 - ISV) In the beginning, God created the universe.
```

The presentation for the translation note becomes a hyphen followed by the value of `transl` in bold font. This presentation of the translation note will continue until you either leave the group where `\translFmt` was used, or you expand `\translFmtDef` to return the presentation to its package default, given above to the right in parentheses.

¶ **`\fetchversestxt`.** When using `\fetchversestxt`, change the default presentation style for the translation note using `\translTxFmt` and `\translTxFmtDef`.

```
\translTxFmt{\<fmt-transl-#1>}
\translTxFmtDef (\translTxFmt{\space#1})
```

These declarations only effect the `\versetxt` command defined by `\fetchversestxt`. Thus,

```
\translTxFmt{ - \textbf{#1}} (Note space after left brace)
\fetchversestxt[from*=ISV]{Gen 1:1}
Gen 1:1 - ISV
```

When using `\fetchversestxt`, a common presentation style is given in the next example:

```
\translTxFmt{ (#1)} (Note space after left brace)
\fetchversestxt[from*=ISV]{Gen 1:1}
\versetxt expands to Gen 1:1 (ISV)
```

The sample file `fetchversestxt.tex` illustrates `\translTxFmt`.

### 3.5. Setting various spaces

- **The space prior to the reference for `citeatend`**

When the option `citeatend=true`, the `\priorRefSpc` command sets the spacing between the passage and the reference.

```
\priorRefSpc{<arg>} (\space)
Other choices for <arg> are \fbFitItIn and \fbFi1lRght
```

Without any declaration, the default argument is `\space` (shown in parentheses); other choices are also listed above for the `<arg>`. The `<arg>` can be some other custom declaration.

**Examples** We illustrate the effects of `\priorRefSpc`.

```
\fetchverses*{Gen 1:1-2} (the default)
```

<sup>1</sup>In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters. (Gen 1:1-2)

```
\priorRefSpc{\fbFi1lRght}\fetchverses*{Gen 1:1-2}
```

<sup>1</sup>In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

(Gen 1:1-2)

```
\priorRefSpc{\fbFitItIn}\afterBookSpc{\nobreak\ }%
```

```
\fetchverses*{Gen 1:1-2}
```

<sup>1</sup>In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

(Gen 1:1-2)

- **Setting the space after the book name**

```
\afterBookSpc{<value>} (\_ )
\resetAfterBookSpc
```

The *value* of the `\afterBookSpc` determines the spacing just after the book name and before the chapter and verse, it is normally a hard space (`\_`). However, you can pass a different value.



**Example** Illustrate the placement of the *value* of `\afterBookSpc`:

```
\afterBookSpc{\textbf{!!}}\fetchverse{Mat 2:1}\resetAfterBookSpc
(Mat!!2:1) Now when Jesus was born in Bethlehem of Judaea in the days of Herod
the king, behold, there came wise men from the east to Jerusalem,
```

The command `\resetAfterBookSpc` resets the change by to its default. In the example, the use of `\resetAfterBookSpc` was not needed since the change occurred within a quote environment. A declaration of `\afterBookSpc{}` removes the space completely. There may be some application to this command.

- **The space following the passage reference**

```
\afterRef{<value>} (\@empty)
```

The `\afterRef` command places its *<value>* immediately following the passage *reference* when the option `citeatend` is *not in effect*:

```
\afterRef{\textbf{!!}}\fetchverses*{1Jn 1:1}
(1Jn 1:1)!!That which was from the beginning, which we have heard, which we have
seen with our eyes, which we have looked upon, and our hands have handled, of the
Word of life;
```

The default value is `\space` obtained by declaring `\afterRef{}` with the empty argument. To get no space after the passage reference and the beginning of the passage use `\afterRef{}`; this latter declaration was used in the example of the `priorref` key of the `fetch` commands.

- **The space between the book and book number**

```
\afterBookSpc{<value>} (\space)
\resetAfterBookSpc
```

The `\afterBookSpc` places *<value>* between the book number and the book, if there is a book number. (E.g., 1 Corinthians).

**Example** Remove the space and use roman numbers:

```
\afterNumSpc{\fetchverse[abbr=none]{1Co 1:1}
(1Corinthians 1:1) Paul, called to be an apostle of Jesus Christ through the will of
God, and Sosthenes our brother,
\afterNumSpc{\fetchverse[abbr=none,roman]{1Co 1:1}
(ICorinthians 1:1) Paul, called to be an apostle of Jesus Christ through the will of
God, and Sosthenes our brother,
```

Of course, the default is to leave a space between the number and the book.

```
\fetchverse[abbr=none,roman]{1Co 1:1}
(I Corinthians 1:1) Paul, called to be an apostle of Jesus Christ through the will of
God, and Sosthenes our brother,
```

### 3.6. Formatting the verse numbers

We define three commands to support the formatting of the verse numbers.

```
\renewcommand\verseFmt{\tex_code_for_#1}
\verseFmtReset
```

The definition of `\verseFmt` determines the formatting of the verse number in the passage; the argument of `\verseFmt (#1)` is the verse number. There is a default definition is

```
\newcommand{\verseFmt}[1]{\verseFmtDef{#1}}
```

where `\verseFmtDef` is the default command to typeset the verse numbers as a superscript. (We don't present its definition in this documentation.)

After redefinition, to return to the default definition, just expand `\verseFmtReset`. Of course, `\verseFmtDef` may be redefined to have a different default definition. (Refer to `fetchbibpes.dtx` for its definition.)

Below are verses showing the default representation of the verse numbers.

(Genesis 1:1-5) <sup>1</sup>In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters. <sup>3</sup>And God said, Let there be light: and there was light. <sup>4</sup>And God saw the light, that it was good: and God divided the light from the darkness. <sup>5</sup>And God called the light Day, and the darkness he called Night. And the evening and the morning were the first day.

The verse numbers appear as superscripts prior to the passages.

However, this formatting can be redefined to obtain the style of numbers you want. In the passage that follows, I've changed the formatting style seen in some Bibles of the multiple passages.

#### GENESIS 1:1-5

- 1 In the beginning God created the heaven and the earth.
- 2 And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.
- 3 And God said, Let there be light: and there was light.
- 4 And God saw the light, that it was good: and God divided the light from the darkness.
- 5 And God called the light Day, and the darkness he called Night. And the evening and the morning were the first day.

```
\begin{quote}
\newlength\verseBoxLength
\settowidth{\verseBoxLength}{\normalsize\normalfont000}
\renewcommand{\verseFmt}[1]{\par\noindent
\makebox[\verseBoxLength][c]{#1}}
```

```
\afterRef{\par\kern3pt}
\fetchverses*[showfirst,delims,caps,
  format={\bfseries\sffamily},abbr=none]{Gen 1:1-5}
\end{quote}
```

It is not necessary in this example for the changes occurred within a group, but to switch back to the default verse number style, we declare `\afterRef{} \verseFmtReset`.

The description of `pcolor=<color>` on [page 17](#) has another example of redefining `\verseFmt`.

**See demo file:** See the demo file `bibpes_tst.tex` for a working version of this example.

Immediately following the verse number is an opportunity to insert an additional command (as a function of the verse number).

```
\afterVerseFmt{\cmd} (\afterVerseFmt{})
```

where `\cmd` is a command taking one argument, the verse number. Consider the following quotation:

**!!Verse 1!!** In the beginning God created the heaven and the earth. **!!Verse 2!!** <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

The verbatim listing of the above is

```
\begin{quote}\def\mrk#1{\textbf{!!Verse~#1!!}}\afterVerseFmt{\mrk}
\fetchverses[showfirst]{Gen 1:1-2}
\end{quote}
```

This is not a very interesting application, but a more interesting use of this command lies ahead. To revert to the default definition execute `\afterVerseFmt{}` with the empty argument.

### 3.7. Dealing with *Strong's Concordance* numbers

When the KJV+ is used, for example, the text is annotated with cross-references to *Strong's Concordance*. In the text file, the references are of the form 'G|H\d+', where `\d+` means one or more natural numbers, 'G' and 'H' refer to the Greek and Hebrew scriptural texts. The references come through in the TXT files as 'G3739' or 'H7225'. When using the WinEdt editor, you can use a regular expression find and replace these references with command versions (e.g., `\G{3739}` or `\H{7225}`):

Greek Strong's	Hebrew Strong's
find: <code>G\(+\\$Numeric\$\)</code>	find: <code>H\(+\\$Numeric\$\)</code>
replace: <code>\\G{\0}</code>	replace: <code>\\H{\0}</code>

**Macro File.** This can be done in either the TXT or DEF file. If you are using WinEdt, you can use the specially written Macro File `bibpes_strongs.edt`, which is found in the `extras` folder. Open the TXT or DEF file (DEF preferred) in WinEdt.

**Procedure.** The procedure for using this Macro File is as follows: (1) Use the menu Macro > Execute Macro Script, browse for `fbpes_strongs.edt` and select it; (2) the Find and Replace dialog box opens with various fields pre-filled; (3) Press 'Replace All', and the regular expression find and replace looks for expressions of the form `Gnnnn` or `Hnnnn`, and replaces them with `\G{nnnn}` and `\H{nnnn}`. The definitions of `\G` and `\H` follow; they may be redefined.

**Example** Below is an example from KJV+, with annotations to *Strong's Concordance*.

```
\fetchverses*[useperiod=false,abbr=short,from*=KJV+]{Rom 1:1}
(Rm 1:1 KJV+) Paul,G3972 a servantG1401 of JesusG2424 Christ,G5547 calledG2822 to be
an apostle,G652 separatedG873 untoG1519 the gospelG2098 of God,G2316
```

### 3.8. The CFG file of the package

`ignorecfg` option Unless the option `ignorecfg` option appears in the option list of `fetchbibpes`, the configuration file `fbpes.cfg` is input at the end of the package.

There is not much to 'configure' in this package, however. The application that was in mind when this feature was created is the redefinition of the `from` key of the `fetch` command options.

```
\defaultBible{<transl>}
\registerBibles{<comma-delimited-list>}
```

The above is the suggest contents of the `fbpes.cfg`: (1) first establish the default Bible, if different than the one already declared; (2) list the Bible translations to be 'registered'. The registered Bibles will be recognized by the `from` key, without having to first pass them through the `\addtoBibles` command ('Adding Bibles' on page 27).

For example, you may routinely use many more books than the minimal ones set up by this package, so in your own CFG file you might define,

```
\defaultBible{NKJV}
\registerBibles{NKJV,KJV,KJV+,ISV,RV,YLT}
```

`\registerBibles` discussed The `\registerBibles` is a command not previously mentioned. Its argument is a comma-delimited list of Bible translations recognized by e-Sword.

A less complicated method that does not modify the definition of the `from` key of the `fbpes` family is to simply use the `\addtoBibles` command in the CFG file. For example, in the CFG you can have,

```
\defaultBible{NKJV}
\addtoBibles{NKJV,RV,YLT}
```

The `fetchbibpes` package already recognizes KJV, KJV+, and ISV, the above declarations adds to this list and makes NKJV the default Bible.

### 3.9. Marginal notes

We introduce a scheme for placing notes in the margins. The `\verseCmts` command is used to define the notes (comments), each note corresponds to a verse.

```
\verseCmts{
  {\comment1} % (first comment of passage)
  {\comment2} % (second comment of passage)
  ...
  {\commentn} % (last comment of passage)
}
```

For example, the following declaration is used in the example below.

```
\verseCmts{%
  {In the beginning God creates\dots} % v1
  {Earth void} % v2
  {Light!} % v3
  {} % v4
  {First day} % v5
}
```

Its argument is a series of tokens (comments enclosed in braces), the first token is the comment for verse 1, then verse 2, and so on. All verses must have a comment, even an empty comment.

The marginal notes are placed in the margin using `\marginpar`. The formatting for the marginal notes is set by `\fbMarParFmt`:

```
\newcommand{\fbMarParFmt}[1]{%
  \marginpar{\footnotesize\raggedright\verseFmt{\vrseNum}#1}}
\fbResetMarParFmt
```

The default definition of `\fbMarParFmt` is seen above. The command may be redefined as needed. (This is done in the demo file `bibpes_tst.tex`.) If `\fbMarParFmt` is redefined, then expanding `\fbResetMarParFmt` returns it to its original definition.

Now to make this all work - that is, to make marginal comments appear - you need two more commands, these are `\fbMarNotesOn` and `\fbMarNotesOff`. The former to turn on marginal notes, the latter to turn switch off again.

```
\fbMarNotesOn \fbMarNotesOff
```

Both make a `\afterVerseFmt` declaration to insert the marginal the marginal notes (`\fbMarNotesOn`) or to turn off the insertion (`\fbMarNotesOff`).

**Example** Begin by expanding `\fbMarNotesOn`, then using the `\verseCmts` declared above, we fetch some verses.

**GENESIS 1:1-5**

<sup>1</sup>In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters. <sup>3</sup>And God said, Let there be light: and there was light. <sup>4</sup>And God saw the light, that it was good: and God divided the light from the darkness. <sup>5</sup>And God called the light Day, and the darkness he called Night. And the evening and the morning were the first day.

<sup>1</sup>In the beginning God creates...

<sup>2</sup>Earth void

<sup>3</sup>Light!

<sup>5</sup>First day

After this passage, we turn everything off with `\fbMarNotesOff\verseCmts{}`, not required if the passage is within a group.

Actually, for this example there is extensive background redefinitions to fit the marginal notes to appear within the margins, as seen above.

**Demo file:** The verbatim listings for this example, as well as for a basic example, are found in the demo file `bibpes_tst.tex`.

**3.10. Retrieving only the verses needed**

One user wants to have *an entire book* (1Jn, for example) and to retrieve only the verses needed. Thus, if

```
\BV(\book\chapter):\verse\transl)\passage\null
```

is one of the required verses, it would be input as usual, otherwise the line is ignored.

useselverses  
option

To obtain selected verses, first specify the `useselverses` option in the option list of `fetchbibpes`. (This option inputs this code segment.) Now the document author needs to specify which of the verses he or she is to use. For that purpose the `\selectedVersesFrom` is defined. The syntax is presented next.

```
\selectedVersesFrom{
  {from=\version},onlyverses={\verses}}
  ...
}
```

Recognized keys: from,  
onlyverses, alt

from, alt,  
onlyversion

The argument for `\selectedVersesFrom` is a series of key-values enclosed in braces. Within the argument, three keys are recognized, `from`, `onlyversion`, and `alt`. For example,

```
\selectedVersesFrom
{
  {onlyverses={1Jn 1:1-5,3:3-8}}
  {from=ISV,onlyverses={Gen 1:1-5}}
  {from=ISV,alt=Alt,onlyverses={Gen 1:1}}
  {onlyverses=Mat}
}
```

It is assumed the required DEF files are listed in the value of the `useverses` option of `fetchbibpes`. Note that the syntax for the value of `onlyverses` is more general; in the 1Jn book, the author intends to use verses 1:1–5 and 3:3–8. Multiple references to the same book are separated by a comma. If the desired book is an alternate translation than the default one, use the `from` key, as above. In the case of `{onlyverses=Mat}`, all verses from the book of Matthew are excluded for no verses are specified.

The book reference can be either the three-letter abbreviation of e-Sword or the acceptable alternate abbreviation, refer to the **e-Sword Abbr** and **Alt\_Abbr** columns of [Table 1](#).

**Demo file:** See the test file `bibpes_sf.tex`. Within this demo file, many finer details are explained and illustrated.

### 3.11. The `\sr` command

The `\sr` command is designed to be used with the `bibleref-parse` package.

```
\sr{<text>}{<passages>}
```

`usesr` option For `\sr` to be defined, the option `usesr` must be specified in the options list of `fetchbibpes`.

`\sr` is a command that takes two arguments, if the first one is non-empty, the `<text>` argument appears in *italics*. The second argument, `<passages>`, are references to one or more books of the Bible and passages within the books. The syntax for the second argument is illustrated by an example:

```
Psa 91:3,14-16;1Co 11; Galatians 5:6
```

The first two books referenced uses the three-letter abbreviation of e-Sword, the last one (Galatians) uses a full book name. (Refer to [Table 1](#)) Multiple passages within the same book are separated by commas, while references of different books are separated by semicolons. There can be a space following a semicolon. This second argument is passed to the `\pibibleverse` command of the `bibleref-parse` package.

**Demo file:** The simple demo file is `bibps_sr.tex`.

### 3.12. The `declareBVs` and `declareBVs*` environment

The `declareBVs` and `declareBVs*` environments may be used anywhere in the document - preamble or body - to declare Biblical verses. Normally, it is used to declare verses not already included by the `useverses=<list>` option, or when there are no `useverses` includes. The option `useverses=none` is appropriate in this last case. The syntax for the content of the `declareBVs` environment is,

```
\begin{declareBVs}
\BV(Gal 1:15 NKJV) But when it pleased God, who separated me from
my mother's womb and called me through His grace,\null
\BV(Gal 1:16 NKJV) to reveal His Son in me, that I might preach Him
among the Gentiles, I did not immediately confer with
flesh and blood,\null
\end{declareBVs}
```

where `\BV` is the low-level command first discussed in [Section 2.2](#) on page 7. The general syntax for `\BV` is,

```
\BV(<verse-ref>)\_<passage>\null
```

where `<verse-ref>` is the specification of the verse referenced. Note that it is enclosed in parentheses, not braces. The general form of `<verse-ref>` is given in the expanded display below.

```
\BV(<book>\_<chapter>):<verse>\_<transl>[_<word>])\_<passage>\null
```

The symbol ‘`_`’ here is a space-character (*not* a hard space). The `<passage>` ends with `\null`, which acts as a delimiter for the `\BV` macro. Following `<transl>` (which corresponds to the value of the `from` key) is an optional `alt <word>` modification, refer to the discussion of the `alt` key on [page 14](#).

Within the `declareBVs` environment, the passages can be allowed to wraparound, unlike when they appear in a DEF file.

Once the passages are declared, they can be retrieved at any point thereafter:

```
\fetchverses*[from=NKJV]{Gal 1:15-16}
```

(Gal 1:15-16) But when it pleased God, who separated me from my mother’s womb and called me through His grace,<sup>16</sup> to reveal His Son in me, that I might preach Him among the Gentiles, I did not immediately confer with flesh and blood,

The `declareBVs` environment is used in the sample files `bibpes_replace.tex`, `bibpes_replaces.tex`, and `bibpes_red.tex`.

The `declareBVs*` environment is the same as `declareBVs` but does not define single and double quote as active characters. Refer to [Section 3.15](#) for more information on quotation marks.

### 3.13. Fetching a substring of a verse

Why can’t copy and paste just be used? A user wants it, I hear and obey. The new code for extracting a substring of a (single) verse is loaded in using the package option `fetchsubtr`. The command tool for identifying the substring or substrings of a verse to be extracted is `\makeverse`. The demonstration file for this feature is `bibpes_substr.tex`.

```
\markverse[name=<name>, from=<transl>,
marks={\bfind1}\efind1}
{\bfind2}\efind2}...{\bfindk}\efindk}
]{<book>\_<ch>:<vrs>}
```

Only *one verse may be marked off* at a time (of course multiple uses of `\markverse` are permitted), but several substrings within that verse may be marked off. There must be an even number of marks. The substrings must be non-overlapping and the order the markers appear in the list of marks is the order they appear in the passage. The command `\markverse` uses the replace feature developed earlier. The value of the name key must be unique throughout all names declared by the `\markverse` command.



The marks ( $\langle bfind \rangle$  and  $\langle efind \rangle$ ) are text (words or phrases) found in the verse. Ideally, the marks *should not contain any quotes* (single or double) unless the verse is defined within the `declareBVs*` environment, where quotes are not active.

Before illustrating, let's have a conceptual example. We begin with

(Gen 1:1) In the beginning God created the heaven and the earth.

Suppose we wanted extract 'In the beginning' and 'the heaven and earth' from this short verse. We use `\markverse` in this way:

```
\markverse[name=Gen1:1-,marks={In the}{God}
{the heaven}{earth.}]{Gen 1:1}
```

When this command line is expanded, the supporting commands of the `replace` key (pages 21-22) are used to mark up this passage, like so:

```
\bMrkFP In the beginning God\eMrkFP created \bMrkFP the heaven and
the earth.\eMrkFP
```

Each pair of marks is then used to scoop up the text string between them. Cool.

Let's try now: The first substring is "In the beginning God"; the second substring is "the heaven and the earth."

The verbatim listing of the previous paragraph is,

```
Let's try now: The first substring is
\fetchverses[alt=Gen1:1-1,enclosewith=\quote]{Gen 1:1};
the second substring is
\fetchverses[alt=Gen1:1-2,enclosewith=\quote]{Gen 1:1}
```

The substrings are accessed through the `fetch` commands by concatenating the value of the  $\langle name \rangle$  key with the index of the substring (first substring=1, second substring=2, and so on) and assigning it as the value of the `alt` key.

The `\markverse` command saves the substrings to the file `\jobname-bv.cut` and inputs them at the top of the file, as a result it may take two compiles to see the changes. From the above `\markverse` declaration, the file `fetchbibpes_man-bv.cut` reads,

```
\begin{declareBVs*}
\BV(Gen 1:1 KJV Gen1:1-1) In the beginning God\null
\BV(Gen 1:1 KJV Gen1:1-2) the heaven and the earth.\null
\end{declareBVs*}
```

The substrings are accessed through the `alt` key of the `fetch` commands. The declaration `\fetchverses[alt=Gen1:1-1,enclosewith=\quote]{Gen 1:1}` retrieves the `Gen 1:1 KJV Gen1:1-1` substring verse. Because the substrings are so defined, the power of the `fetch` verse commands can be brought to bare.

If a substring is not found (that is, for various reasons the  $\langle bfind \rangle$  and/or  $\langle efind \rangle$  marks are not found), the an index string is still defined and expands to an alert message. For example, we declare

```
\markverse[name=errorExmp],marks={In the}{God}
{the Heaven}{earth.}]{Gen 1:1}
```

You'll note the fourth mark is 'the Heaven' not 'the heaven' (the search is case sensitive). What are the results:

- `\fetchverse[alt=errorExmpl1]{Gen 1:1}`:  
(Gen 1:1) In the beginning God
- `\fetchverse[alt=errorExmpl2]{Gen 1:1}`:  
(Gen 1:1) ??(alt=errorExmpl2: substring not identified)??

As predicted, search for the second substring failed, and we are so informed.

**Contiguous substrings.** One useful feature of the marks key of `\markverse` is the odd numbered markers may be the empty string. If one is empty, the search continues from where the previous search stopped. The first example could have been posed as follows:

```
\markverse[name=Gen1:1-,marks={}{God}
{the heaven}{earth.}]{Gen 1:1}
```

The first marker is empty ({}), so the search begins where the previous one left off, but this is the first search so the search begins at the beginning of the verse. A more informative example would be the following. Declare,

```
\begin{declareBVs*}
\BV(Mat 6:31 NKJV) ‘Therefore do not worry, saying,
‘What shall we eat?’ or ‘What shall we drink?’ or
‘What shall we wear?’\null
\end{declareBVs*}
```

Note the presence of quote marks, hence, we declare this verse using `declareBVs*`. We emit our `\markverse` command:

```
\markverse[name=Mat6:31-,from=NKJV,
marks={}{eat?'}{drink?'}{wear?'}]{Mat 6:31}
```

Now, let's read out the substrings:

- `\fetchverse[alt=Mat6:31-1,from=NKJV]{Mat 6:31}`:  
(Mat 6:31) “Therefore do not worry, saying, ‘What shall we eat?’
- `\fetchverse[alt=Mat6:31-2,from=NKJV]{Mat 6:31}`:  
(Mat 6:31) or ‘What shall we drink?’
- `\fetchverse[alt=Mat6:31-3,from=NKJV]{Mat 6:31}`:  
(Mat 6:31) or ‘What shall we wear?’

Contiguous substrings as promised!

In this example, `from=NKJV` was used in both the `\markverse` and `\fetchverse` commands to identify the translation, this is required if the passage comes from a translation other than the default translation (KJV in this document).

**Warning!** Because the code for the `replace` feature is used, the entire verse is searched for the find text and replaced, possibly multiple times! Multiple replacements in this context is never wanted, I would think. Therefore, the markers `<bfind>` and `<efind>` should *occur once and only once* in the verse. For example the line

```
\markverse[name=multihits,from=NKJV,
marks={'What'}{wear?}]{Mat 6:31}
```

produces the following `\BV` definition in the `fetchbibpes_man-bv.cut` file:

```
\BV(Mat 6:31 NKJV multihits1) 'What shall we eat?' or \bMrkFP
'What shall we drink?' or \bMrkFP 'What shall we wear?'
```

Oops, multiple `\bMrkFP` markers linger where they shouldn't.

### 3.14. Some extended quote environment

At the mild prompting of one of the (only two known) users, three environments are also defined for quoting long passages. These are minor modifications of the  $\text{\LaTeX}$  environments `quote`, `verse`, and `quotation`.

```
\begin{fpquote}[<options>]
  <content>
\end{fpquote}

\begin{fpverse}[<options>]
  <content>
\end{fpverse}

\begin{fpquotation}[<options>]
  <content>
\end{fpquotation}
```

When any of the three environments `fpquote`, `fpverse`, and `fpquotation` are used without the optional argument, they behave exactly as the  $\text{\LaTeX}$  core counterparts do. The optional argument recognizes three keys `leftmargin`, `rightmargin`, and `itemindent`; these `\leftmargin`, `\rightmargin`, and `\itemindent`, respectively. A few examples suffice.

- `fpquote` only obeys `leftmargin`, `rightmargin`.

```
\begin{fpquote}[rightmargin=0pt]
```

(Gen 1:1-2) In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

(Gen 1:3-4) And God said, Let there be light: and there was light. <sup>4</sup>And God saw the light, that it was good: and God divided the light from the darkness.

This is perhaps the most common use for this environment, to indent on the left, but not on the right. The key `itemindent` is not obeyed for the `fpquote` environment.

- The verse gives an outdent effect.

```
\begin{fpverse}[rightmargin=0pt]
```

(Gen 1:1-2) In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

We can adjust the amount of outdent with `itemindent`.

```
\begin{fpverse}[itemindent=3em,rightmargin=0pt]
```

(Gen 1:1-2) In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

(Gen 1:3-4) And God said, Let there be light: and there was light. <sup>4</sup>And God saw the light, that it was good: and God divided the light from the darkness.

The default for `itemindent` is 1.5em.

- The `fpquotation` environment is like the `fpverse` but each paragraph is indented by an amount of `\itemindent` (set by `itemindent`).

```
\begin{fpquotation}[rightmargin=0pt]
```

(Gen 1:1-2) In the beginning God created the heaven and the earth. <sup>2</sup>And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.

(Gen 1:3-4) And God said, Let there be light: and there was light. <sup>4</sup>And God saw the light, that it was good: and God divided the light from the darkness.

The default value for `itemindent` is 1.5em.

### 3.15. Troubles with quotation marks and apostrophes

Some Bible translations use more quotes (both single and double) than others. The English Standard Version (ESV) uses an overabundance of such. Quotation marks are not only used to quote the speaker, but the single quote (') is often used as an apostrophe to indicate possession, as in "God's grace." (' `God's grace. '). 'Most' Bible translations, as copied from e-Sword, use double quotes (") and single quotes (') for quoting passages. At least one translation, Young's Literal Translation (YLT), uses single quotes in the same way as does  $\LaTeX$ ; for example, *according to Gen 1:3*, ??(Gen 1:3 YLT)??. If you look in the DEF file you'll see that this passage reads, in part, 'Let light be;', which is good for user's of  $\LaTeX$ . But there are problems.

`fetchbibpes` tries to match pairs of double quotation marks ("*text*") and single quotes ('*text*'). It maintains two count registers `\fp@dqcnt` and `\fp@sqcnt`. When a double quote is encountered and `\fp@dqcnt` is even, "" is replaced by ``; similarly, when `\fp@dqcnt` is odd, "" is replaced by ' '. For the single quote it's the same thing. If a given passage has balanced quotes we don't have a problem, for single quote used as an apostrophe, usually, we have to make adjustment so the typesetting is correct.

For this purpose, the commands `\bDQ`, `\eDQ`, `\bSQ`, and `\eSQ` are defined. `\bDQ` and `\bSQ`, `\eDQ`, `\eSQ`

`\bSQ` set their respective counters to zero (0), which is an even number, while `\eDQ` and `\eSQ` set their respective counter to one (1), which is odd.

**Example** Verses 3 and 5 use an apostrophe, we use the following commands to get the single quite (apostrophe) correct. This passage comes from Matthew 7:1-7 (*ESV*):

<sup>1</sup>“Judge not, that you be not judged. <sup>2</sup>For with the judgment you pronounce you will be judged, and with the measure you use it will be measured to you. <sup>3</sup>Why do you see the speck that is in your brother’s eye, but do not notice the log that is in your own eye? <sup>4</sup>Or how can you say to your brother, ‘Let me take the speck out of your eye,’ when there is the log in your own eye? <sup>5</sup>You hypocrite, first take the log out of your own eye, and then you will see clearly to take the speck out of your brother’s eye.” <sup>6</sup>“Do not give dogs what is holy, and do not throw your pearls before pigs, lest they trample them underfoot and turn to attack you.” <sup>7</sup>“Ask, and it will be given to you; seek, and you will find; knock, and it will be opened to you.

The verbatim listing is

```
\eSQ\fetchverses[showfirst,from=ESV]{Mat 7:1-4}
\eSQ\fetchverses[showfirst,from=ESV]{Mat 7:5-7}
```

It was necessary to break the passage up so the two apostrophe were included in separate fetches; prior to each fetch we declared `\eSQ`, which makes `\fp@sqcnt` odd, and so the next single quote (') is typeset as ' rather than '. Notice that in verse 4 there are matching single quotes, following the apostrophe of verse 3, `\fp@sqcnt` should be even, so the matched single quotes are handled correctly. Again, in verse 5 there is an apostrophe, we use `\eSQ` to make the counter odd so the apostrophe is typeset correctly.

**Example** In the next passage (Mat 7:13-20 (*ESV*)) there are several troubling quotation marks. In verse 13 a left double quote opens a quotation, but it is never closed; verse 15 starts another left double quote. Also, there is an apostrophe in verse 15 that needs to be displayed correctly.

<sup>13</sup>“Enter by the narrow gate. For the gate is wide and the way is easy that leads to destruction, and those who enter by it are many. <sup>14</sup>For the gate is narrow and the way is hard that leads to life, and those who find it are few. <sup>15</sup>“Beware of false prophets, who come to you in sheep’s clothing but inwardly are ravenous wolves. <sup>16</sup>You will recognize them by their fruits. Are grapes gathered from thornbushes, or figs from thistles? <sup>17</sup>So, every healthy tree bears good fruit, but the diseased tree bears bad fruit. <sup>18</sup>A healthy tree cannot bear bad fruit, nor can a diseased tree bear good fruit. <sup>19</sup>Every tree that does not bear good fruit is cut down and thrown into the fire. <sup>20</sup>Thus you will recognize them by their fruits.”

The verbatim listing is

```
\fetchverses[showfirst,from=ESV]{Mat 7:13-14}
\eSQ\bDQ\fetchverses[showfirst,from=ESV]{Mat 7:15-20}
```

The double quote of verse 13 opens correctly. Beginning with verse 15, we make the adjustments `\eSQ\bDQ`.

The King James Version does not have many quote marks, if any, but does have apostrophes; one runs into challenges with other translations. The rule here is, when quoting from a translation that uses quotes and apostrophes, keep an eagle eye out for them.

#### 4. Combining bibpes and fetchbibpes

Though the two packages were originally designed to work separately, but in concert with each other, it is possible to combine the two packages in one source file, yet keep the functionalities of the two separated.

To do this, use the `spdef` package, which comes bundled with the `aeb_mobile` package (<http://ctan.org/pkg/aeb-mobile>). The technique is roughly outlined below.

```

1 \RequirePackage[use=buildddefs]{spdef}
2 \ifbuildddefs
3 \documentclass{article}
4 \usepackage[<opts>]{bibpes}
5 \bibpesBody\fi
6 %
7 % Begin the fetchbibpes portion of the document
8 %
9 \documentclass{article}
10 \usepackage[<opts>]{fetchbibpes}
11 \begin{document}
12   <document content, containing fetch verse commands>
13 \end{document}
```

Line (1) introduces `spdef` using `\RequirePackage`. The option `use=buildddefs` defines a Boolean switch `\ifbuildddefs` and sets it to `true`. When the source document is `latex`ed in this state, only the lines (2)–(5) (the `bibpes` portion of the document) are compiled. When the source document is `latex`ed with option `!use=buildddefs` (note the exclamation mark), the Boolean switch `\ifbuildddefs` is once again created but this time it is set to `false`; consequently, lines (2)–(5) are scanned over (ignored) and lines (9)–(13) (the `fetchbibpes` portion of the document) are compiled.

**Demo file:** `combpkgs.tex`

#### 5. My retirement

Now, I simply must get back to it. ~~DS~~