The **scripture** package

David Purton*

2025/01/02 v2.1

Abstract

The scripture package provides a set of macros for typesetting quotations from the Bible. It provides many features commonly seen in Bibles such as dropped text for chapter numbers, superscripts for verse numbers, indented lines for poetry sections, narrow sections and hanging paragraphs. A reference for the quotation can optionally be added.

Contents

---

*Email: dcpurton@marshwiggle.net

1   Introduction

Bibles tend to use a number of standard features when laying out chapters and verses. The main goal for the scripture package is to be able to quickly and easily input text and have it output nicely with minimal manual adjustment.

The package supports dropped text for chapter numbers, superscripts for verse numbers, appropriately indented lines for the first and second halves of poetry, flush right sections, centred sections, narrow sections and hanging sections.

The package could be used for typesetting an entire Bible, but it is mainly envisioned that it will be used for producing Scripture quotations. The passage reference can be included on the last line if there is space or failing that on the following line.

1.1   Bug reports and feature requests

Bug reports and feature requests can be made at the scripture package GitHub repository. See <https://github.com/dcpurton/scripture>.

1.2 Limitations

The scripture package makes extensive use of LaTeX `list` environments and the `\parshape` command. This means there are some limitations. In particular, you should not:

- Nest the `scripture` environment or any of its internal environments.

- Modify the `\parshape` within a `scripture` environment.

- Include any other `list` based environment within a `scripture` environment.

- Include any `\section` (or similar) functions within the `scripture` environment. A simple `\heading` function is provided instead.

The `poetry` environment uses `\obeylines`. This means that it can not be placed in the argument of a macro. If you want to use the `poetry` environment in a `beamer` slide, you must use the `fragile` option for the `beamer frame` environment.

Some attempt is made to encourage appropriate page breaks, especially just before a reference is added to a quotation, but this can still sometimes fail and a reference will end up on the following page. Usually a little more stretchable space on the page before the incorrect break is enough to obtain acceptable output.

At least version 2022-11-01 of the LaTeX format is required.

2 Documentation

2.1 Basic usage

Scripture quotations must be set within a `scripture` environment. A simple quotation might be produced as follows:

```
\begin{scripture}[Isaiah 56:1--2][version=NIVUK]
  \ch{56}This is what the \name{Lord} says:
  \begin{poetry}
    `Maintain justice
    and do what is right,

    for my salvation is close at hand
    and my righteousness will soon be revealed.

    \vs{2}Blessed is the one who does this~---
    the person who holds it fast,

    who keeps the Sabbath without desecrating it,
    and keeps their hands from doing any evil.'
  \end{poetry}
\end{scripture}
```

56 This is what the Lord says:

'Maintain justice
  and do what is right,
for my salvation is close at hand
  and my righteousness will soon
    be revealed.
²Blessed is the one who does this –
  the person who holds it fast,
who keeps the Sabbath without
    desecrating it,
  and keeps their hands from
    doing any evil.'
Isaiah 56:1–2 (NIVUK)

2.2 Macros for typesetting scripture quotations

2.2.1 Functions

`\added`

`\added{⟨words⟩}`

New: 2022-11-15 Some Bible translations mark out words added for clarity not found in the original language (e.g., in italics). This can be done using the `\added` function. The font and format can be set using the `added/font` and `added/format` options.

```
\begin{scripture}[John 1:18][verse/font=\bfseries,
    verse/format=#1, verse/sep=0.5em, version=KJV]
  \vs{18}No man hath seen God at any time, the
  only begotten Son, which is in the bosom of the
  Father, he hath declared \added{him}.
\end{scripture}
```

18 No man hath seen God at any time, the only begotten Son, which is in the bosom of the Father, he hath declared *him*.   **John 1:18 (KJV)**

`\ch`   `\ch[⟨letter⟩]{⟨chapter number⟩}`
`\ch*` `\ch*[⟨letter⟩]{⟨chapter number⟩}`

Format a chapter number. By default these are formatted using drop text taking up two lines and the text is separated from the chapter number by a horizontal space of 0.5 em. The drop text is aligned to the top of ⟨`letter`⟩ which defaults to the value of `chapter/xchar`, which in turn defaults to a capital X in the default font.

If the `chapter/drop` option is `true` (default), then this function attempts to add sufficient space for the dropped chapter when its paragraph contains only one line. The starred version suppresses this behaviour.

`\ch` **example**

```
\begin{scripture}[Exodus 20:1--2]
  \ch{20}And God spoke all these words:
  \begin{hanging}
    \noindent\vs{2}`I am the \name{Lord}
    your God, who brought you out of Egypt,
    out of the land of slavery.
  \end{hanging}
\end{scripture}
```

`\ch*` **example**

```
\begin{scripture}[Exodus 20:1--2]
  \ch*{20}And God spoke all these words:
  \begin{hanging}
    \nohang\vs{2}`I am the \name{Lord} your
    God, who brought you out of Egypt, out of
    the land of slavery.
  \end{hanging}
\end{scripture}
```

`\ch` **example**

20 And God spoke all these words:

²'I am the LORD your God, who brought you out of Egypt, out of the land of slavery.

Exodus 20:1–2

`\ch*` **example**

20 And God spoke all these words:

²'I am the LORD your God, who brought you out of Egypt, out of the land of slavery.

Exodus 20:1–2

`\extraskip`   `\extraskip`

Add a small vertical space of `extraskip` between a paragraph. This function attempts to add additional space to avoid a clash with a drop chapter if required.

```
\begin{scripture}[Colossians 1:2]
  \vs{2}To God's holy people in Colossae, the
  faithful brothers and sisters in Christ:

  \extraskip

  Grace and peace to you from God our Father.
\end{scripture}
```

²To God's holy people in Colossae, the faithful brothers and sisters in Christ:

Grace and peace to you from God our Father.   **Colossians 1:2**

`\heading`	`\heading{⟨section heading⟩}`

The `\heading` function provides a simple section heading command for use within the `scripture` environment. Only one level is heading is provided. Further levels should be placed outside of the `scripture` environment. Additionally, the `\heading` function may not be used within a `scripture` inner environment.

The settings of headings can be controlled using the following options:

- `heading/aboveskip`

- `heading/afterindent`

- `heading/align`

- `heading/belowskip`

- `heading/font`

- `heading/format`

- `heading/hide`

- `heading/show`

Notes:

- `\section` and related functions are not supported within the `scripture` environment.

- The `\heading` function is ignored when the `inline` option is `true`.

```
\begin{scripture}[Luke 21:5-6]
  \heading{Signs of the End of the Age}

  \vs{5}Some of his disciples were remarking
  about how the temple was adorned with beautiful
  stones and with gifts dedicated to God. But
  Jesus said, \vs{6}``As for what you see here,
  the time will come when not one stone will be
  left on another; every one of them will be
  thrown down.''
\end{scripture}
```

Signs of the End of the Age

[5]Some of his disciples were remarking about how the temple was adorned with beautiful stones and with gifts dedicated to God. But Jesus said, [6]"As for what you see here, the time will come when not one stone will be left on another; every one of them will be thrown down." **Luke 21:5–6**

`\name`	`\name{⟨divine name⟩}`
`\LORD`	`\LORD`
`\GOD`	`\GOD`

Modern Bible translations frequently represent God's personal divine name using a small capitals typeface. This can be done using the `\name` function. The font and format can be set using the `name/font` and `name/format` options. The `\LORD` and `\GOD` macros equivalent to `\name{Lord}` and `\name{God}` respectively.

Note: Watch out that trailing spaces are not eaten by `\LORD` and `\GOD`. Using `\name` is safer and semantically better.

```
\begin{scripture}[Genesis 15:1--2][version=ESV]
  \ch{15}After these things the word of the
  \name{Lord} came to Abram in a vision: ``Fear
  not, Abram, I am your shield; your reward shall
  be very great.'' \vs{2}But Abram said, ``O Lord
  \name{God}, what will you give me, for I
  continue childless, and the heir of my house is
  Eliezer of Damascus?''
\end{scripture}
```

15 After these things the word of the LORD came to Abram in a vision: "Fear not, Abram, I am your shield; your reward shall be very great." ²But Abram said, "O Lord GOD, what will you give me, for I continue childless, and the heir of my house is Eliezer of Damascus?"

Genesis 15:1–2 (ESV)

\nofirstverse \nofirstverse

When `verse/first` is `true`, this causes the next `\vs` to print normally, rather than using the format specified by `verse/firstformat`. You might do this if your quote begins half way through a paragraph and the first verse should be printed normally.

\nohang \nohang

In a `hanging` environment, this function indents the first line of the paragraph a length of `hanging/hang` + `parindent`. Remaining lines continue to be indented by a length of `hanging/hang`.

```
\begin{scripture}[Genesis 1:10--11]
  \begin{hanging}
    \noindent\vs{10}God called the dry ground
    `land', and the gathered waters he called
    `seas'. And God saw that it was good.

    \nohang\vs{11}Then God said, `Let the land
    produce vegetation: seed-bearing plants and
    trees on the land that bear fruit with seed
    in it, according to their various kinds.'
    And it was so.
  \end{hanging}
\end{scripture}
```

¹⁰God called the dry ground 'land', and the gathered waters he called 'seas'. And God saw that it was good.

¹¹Then God said, 'Let the land produce vegetation: seed-bearing plants and trees on the land that bear fruit with seed in it, according to their various kinds.' And it was so.

Genesis 1:10–11

\nopilcrow \nopilcrow

When `pilcrow` is `true`, this causes the next pilcrow that would normally by printed to be suppressed. You might do this if your quote begins half way through a paragraph and so should not be marked with a pilcrow.

\redletteron \redletteron
\redletteroff \redletteroff

Use `\redletteron` and `\redletteroff` to mark the beginning and end of sections of text that should be attributed to Jesus speaking. Some attempt is made to ensure that the correct colour continues after an inner environment if `\redletteron` or `\redletteroff` occurs inside the inner environment. If you experience problems with unexpected colour, try to ensure that matching `\redletteron` and `\redletteroff` marks occur at the same group level. The colour of the text can be set using the `redletter/colour` option. These commands only have an effect when the `redletter` option is `true`.

```
\begin{scripture}[Matthew 21:16--17][redletter]
  \vs{16}`Do you hear what these children are saying?'
  they asked him.

  \redletteron `Yes,' \redletteroff replied Jesus,
  \redletteron `have you
  never read,
  \begin{poetry}
    `\thinspace ``From the lips of children and infants
    you, Lord, have called forth your praise''?'
  \end{poetry}\redletteroff

  \vs{17}And he left them and went out of the city to
  Bethany, where he spent the night.
\end{scripture}
```

[16] 'Do you hear what these children are saying?' they asked him.
'Yes,' replied Jesus, 'have you never read,

' "From the lips of children and infants you, Lord, have called forth your praise"?'

[17] And he left them and went out of the city to Bethany, where he spent the night.
Matthew 21:16–17

\scripturecurrentchapter
\scripturecurrentverse

New: 2022-08-07

\scripturecurrentchapter
\scripturecurrentverse

\scripturecurrentchapter and \scripturecurrentverse globally hold the most recent values parsed to \ch and \vs. They can initialised with \renewcommand* if required and could be used in hooks, for example.

\scripturesetup

\scripturesetup{⟨*options*⟩}

Set default format for quotations. See section 2.4 for supported formatting ⟨options⟩.

\scripturestyle
\scripturestyle*

New: 2023-12-27

\scripturestyle{⟨*name*⟩}{⟨*options*⟩}
\scripturestyle*{⟨*name*⟩}{⟨*options*⟩}

Set ⟨options⟩ which are saved as style with ⟨name⟩. These can later be set with the style option. The starred version appends ⟨options⟩ to an existing style rather than overwriting all options.

\selah

New: 2022-07-29

\selah

In a poetry environment, this function inserts *Selah* at the end of the current line or the end of the following line if there is insufficient room. See options selah/font, selah/format, selah/sep and selah/text for options controlling the output.

```
\begin{scripture}[Psalm 3:2][version=NIV84]
  \begin{poetry}
    \vs{2}Many are saying of me,
    ``God will not deliver him.''\selah
  \end{poetry}
\end{scripture}
```

[2] Many are saying of me,
"God will not deliver him."
Selah
Psalm 3:2 (NIV84)

\textright

New: 2022-08-30

\textright{⟨*text*⟩}

This function inserts ⟨text⟩ at the end of the current line or the end of the following line if there is insufficient room. The textright/sep option controls the minimum allowed space before the ⟨text⟩ to be right aligned.

```
\begin{scripture}[Jeremiah 2:29]
  \begin{poetry}
    \vs{5}`Why do you bring charges against me?
    You have all rebelled against me,'
    \textright{declares the \name{Lord}.}
  \end{poetry}
\end{scripture}
```

> [5] 'Why do you bring charges against me?
> You have all rebelled against me,'
>   declares the LORD.
>   **Jeremiah 2:29**

\textscripture   `\textscripture[⟨reference⟩][⟨options⟩]{⟨quotation⟩}`

New: 2023-01-07   **\textscripture** sets an inline scripture quotation. The quotation is surrounded be the values of `inline/begin` and `inline/end` which default to `` `` `` and `''`, respectively. This function also sets the `compact` option and places the ⟨*reference*⟩ inline. The reference format can be controlled using the following options:

- `inline/reference/format`

- `inline/reference/sep`

- `inline/version/delim`

- `inline/version/format`

Other ⟨*reference*⟩ and ⟨*version*⟩ options are inherited.

    This function is equivalent to using the `scripture` environment with the `inline` option.

    Warning: `\par` is redefined to `\relax` inside the `\textscripture` function. If you have anything other than expected macros and environments inside your `\textscripture` function TeX will likely get stuck in an infinite loop.

```
A famous verse in the Bible is:
\textscripture[John 3:16][verse/hide]{\vs{16}For
God so loved the world that he gave his one and
only Son, that whoever believes in him shall not
perish but have eternal life.}.
```

> A famous verse in the Bible is: "For God so loved the world that he gave his one and only Son, that whoever believes in him shall not perish but have eternal life." **(John 3:16)**.

\vs   `\vs{⟨verse number⟩}`

Format a verse number. By default these are formatted using a superscript.

2.2.2   Environments

All scripture quotations must be set within a `scripture` environment. The rest of the environments documented below are only defined with the `scripture` environment. In most cases there is a default small ⟨*aboveskip*⟩ and ⟨*belowskip*⟩ inserted above and below these internal environments, but these are ignored if the environment is at the start or end of a `scripture` environment.

---

center   `\begin{center}[⟨options⟩]`
    `⟨centred text⟩`
`\end{center}`

Environment for typesetting centred sections of scripture quotations.

See section 2.4.1 for supported formatting ⟨options⟩.

```
\begin{scripture}[Daniel 5:25]
  \vs{25}`This is the inscription that was
  written:
  \begin{center}
    \textsc{mene, mene, tekel, parsin}
  \end{center}
\end{scripture}
```

²⁵‘This is the inscription that was written:

MENE, MENE, TEKEL, PARSIN

Daniel 5:25

---

flushright

```
\begin{flushright}[⟨options⟩]
  ⟨right aligned text⟩
\end{flushright}
```

Environment for typesetting right aligned sections of scripture quotations.

See section 2.4.2 for supported formatting ⟨options⟩.

```
\begin{scripture}[Isaiah 40:5]
  \begin{poetry}[belowskip = 0pt]
    \vs{5}And the glory of the \name{Lord}\ldots
    and all people will see it together.
  \end{poetry}

  \begin{flushright}[aboveskip = 0pt]
    For the mouth of the \name{Lord} \\
    has spoken.'
  \end{flushright}
\end{scripture}
```

⁵And the glory of the LORD…
and all people will see it
together.
For the mouth of the LORD
has spoken.'

Isaiah 40:5

---

hanging

```
\begin{hanging}[⟨options⟩]
  ⟨hanging text⟩
\end{hanging}
```

Environment for typesetting hanging sections of scripture quotations.

See section 2.4.3 for supported formatting ⟨options⟩.

```
\begin{scripture}[Genesis 1:3--5]
  \begin{hanging}
    \vs{3}And God said, `Let there be light,' and
    there was light. \vs{4}God saw that the light
    was good, and he separated the light from the
    darkness. \vs{5}God called the light `day',
    and the darkness he called `night'. And there
    was evening, and there was morn\-ing---the
    first day.
  \end{hanging}
\end{scripture}
```

³And God said, 'Let there be light,'
and there was light. ⁴God
saw that the light was good,
and he separated the light from
the darkness. ⁵God called the
light 'day', and the darkness he
called 'night'. And there was
evening, and there was morn-
ing—the first day.

Genesis 1:3–5

midparachap `\begin{midparachap}[⟨options⟩]`
    ⟨*text containing a chapter break*⟩
`\end{midparachap}`

Environment for typesetting paragraph with a chapter break within it.

    See section 2.4.4 for supported formatting ⟨`options`⟩.

```
\begin{scripture}[1 Corinthians 10:31--11:1]
  \begin{midparachap}
    \vs{31}So whether you eat or drink or
    whatever you do, do it all for the glory of
    God. \vs{32}Do not cause anyone to stumble,
    whether Jews, Greeks or the church of
    God---\vs{33}even as I try to please everyone
    in every way. For I am not seeking my own
    good but the good of many, so that they may
    be saved. \ch{11}Follow my example, as I
    follow the example of Christ.
  \end{midparachap}
\end{scripture}
```

> [31] So whether you eat or drink or whatever you do, do it all for the glory of God. [32] Do not cause anyone to stumble, whether Jews, Greeks or the church of God—[33] even as I try to please everyone in every way. For I am not seeking my own good but the good of many, so that they may be **11** saved. [1] Follow my example, as I follow the example of Christ.
>
> **1 Corinthians 10:31–11:1**

narrow `\begin{narrow}[⟨options⟩]`
    ⟨*quotation*⟩
`\end{narrow}`

Environment for typesetting narrow sections of scripture quotations. If there is no blank line before the start of a `narrow` environment the indent will be suppressed.

    See section 2.4.5 for supported formatting ⟨`options`⟩.

```
\begin{scripture}[2 Kings 18:18--19]
  \vs{18}They called for the king; and Eliakim
  son of Hilkiah the palace administrator, Shebna
  the secretary, and Joah son of Asaph the
  recorder went out to them.

  \vs{19}The field commander said to them, `Tell
  Hezekiah:

  \begin{narrow}
    `\thinspace``This is what the great king, the
    king of Assyria, says: on what are you basing
    this confidence of yours?
  \end{narrow}
\end{scripture}
```

> [18] They called for the king; and Eliakim son of Hilkiah the palace administrator, Shebna the secretary, and Joah son of Asaph the recorder went out to them. [19] The field commander said to them, 'Tell Hezekiah:
>
>   '"This is what the great king, the king of Assyria, says: on what are you basing this confidence of yours?   **2 Kings 18:18–19**

poetry `\begin{poetry}[⟨options⟩]`
    ⟨*first half of verse*⟩
    ⟨*second half of verse*⟩

    ⟨*first half of verse*⟩
    ⟨*second half of verse*⟩
`\end{poetry}`

Environment for typesetting poetry sections of scripture quotations.

Poetry sections of the Bible often have a first half and a parallel second half which is indented a little. These should be set on separate consecutive lines. Leaving a blank line indicates that the next line should return to the first half indent.

See section 2.4.6 for supported formatting ⟨options⟩.

```
\begin{scripture}[Isaiah 1:2]
  \begin{poetry}
    \vs{2}Hear me, you heavens! Listen, earth!
    For the \name{Lord} has spoken:

    `I reared children and brought them up,
    but they have rebelled against me.
  \end{poetry}
\end{scripture}
```

> ²Hear me, you heavens! Listen, earth!
> For the Lᴏʀᴅ has spoken:
> 'I reared children and brought them up,
> but they have rebelled against me.   **Isaiah 1:2**

scripture \begin{scripture}[⟨*reference*⟩][⟨*options*⟩]
    ⟨*quotation*⟩
\end{scripture}

Main environment for typesetting scripture quotations.

The optional ⟨reference⟩ is printed at the end of the quotation. See section 2.4 for supported formatting ⟨options⟩.

```
\begin{scripture}[John 3:16][version=NIV]
  \vs{16}For God so loved the world that he gave
  his one and only Son, that whoever believes in
  him shall not perish but have eternal life.
\end{scripture}
```

> ¹⁶For God so loved the world that he gave his one and only Son, that whoever believes in him shall not perish but have eternal life.
> **John 3:16 (NIV)**

2.3 Provided hooks

scripture/chap/before
scripture/chap/after
scripture/verse/before
scripture/verse/after

New: 2022-08-07

The following public hooks are available:

scripture/chap/before This hook is executed immediately before the `chapter/font` is set for the chapter.

scripture/chap/after This hook is executed immediately after the `chapter/sep` space is inserted.

scripture/verse/before This hook is executed immediately before the `verse/font` is set for the verse.

scripture/verse/after This hook is executed immediately after the `verse/sep` space is inserted.

The following example illustrates how these hooks could be used to place the first and last verse on the page into the header.

```
\documentclass{article}
\usepackage{scripture}
\AddToHook{scripture/chap/after}{%
  \markboth
    {Mark \scripturecurrentchapter
        :\scripturecurrentverse}
    {Mark \scripturecurrentchapter
        :\scripturecurrentverse}}
\AddToHook{scripture/verse/after}{%
  \markboth
    {Mark \scripturecurrentchapter
        :\scripturecurrentverse}
    {Mark \scripturecurrentchapter
        :\scripturecurrentverse}}
\renewcommand*{\scripturecurrentchapter}{1}
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\lhead{\textbf{\rightmark}}
\rhead{\textbf{\leftmark} \textbar\ \thepage}
\begin{document}
\begin{scripture}[][noindent]
  \vs{45}Instead he went out and began to talk
  freely, spreading the news. As a result, Jesus
  could no longer enter a town openly but stayed
  outside in lonely places. Yet the people still
  came to him from everywhere.

  \extraskip

  \ch{2}A few days later, when Jesus again
  entered Capernaum, the people heard that he had
  come home.
\end{scripture}
\end{document}
```

Mark 1:45 ────── **Mark 2:1** | 1

[45] Instead he went out and began to
talk freely, spreading the news. As
a result, Jesus could no longer enter
a town openly but stayed outside in
lonely places. Yet the people still
came to him from everywhere.

2 A few days later, when Jesus
again entered Capernaum, the
people heard that he had come home.

2.4 Options

Default formatting ⟨options⟩ can be set using \usepackage[⟨options⟩]{scripture} or
\scripturesetup{⟨options⟩}. Local formatting ⟨options⟩ can be set by passing them
to the various environments described above.

---

aboveskip aboveskip = ⟨*skip*⟩ Default: 0pt

⟨skip⟩ inserted above the scripture environment.

---

added/font added/font = ⟨*font*⟩ Default: *Empty*

New: 2022-11-15 ⟨font⟩ used for the \added function.

---

added/format added/format = ⟨*macro*⟩ Default: \emph{#1}

New: 2022-11-15 ⟨macro⟩ used to format the words that should be highlighted as added by the translators
using the \added function. #1 contains the text of the argument to \added.

---

after after = ⟨*token list*⟩ Default: *Empty*

New: 2024-06-01 Execute ⟨token list⟩ at the end of the scripture environment.

| **before** | before = ⟨*token list*⟩ | Default: *Empty* |

New: 2024-06-01 Execute ⟨token list⟩ at the beginning of the **scripture** environment.

| **belowskip** | belowskip = ⟨*skip*⟩ | Default: 0pt |

⟨skip⟩ inserted below the **scripture** environment.

| **chapter/align** | chapter/align = left | right | Default: left |

New: 2024-12-30 When **chapter/align** is set to **right** a chapter will overlap to the left when **chapter/nodrop** is set.

See also **chapter/valign**.

| **chapter/colour** | chapter/colour = ⟨*colour expression*⟩ | Default: . |
| **chapter/color** | | |

New: 2022-09-07 The l3color ⟨colour expression⟩ to be used for chapter numbers. By default the current colour is inherited. **chapter/color** is an alias for **chapter/colour**.

| **chapter/drop** | chapter/drop = ⟨*boolean*⟩ | Default: true |

Whether the chapter should be set as a drop letter taking up two lines or as a normal letter taking up one line. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present. This option is the opposite of **chapter/nodrop**.

| **chapter/font** | chapter/font = ⟨*font*⟩ | Default: \bfseries |

⟨font⟩ used for the **\ch** function.

| **chapter/format** | chapter/format = ⟨*macro*⟩ | Default: #1 |

New: 2022-07-28 ⟨macro⟩ used to format the **chapter** for a **scripture** quotation. **#1** contains the text of the **chapter**.

| **chapter/hide** | chapter/hide = ⟨*boolean*⟩ | Default: false |

New: 2022-08-15 Whether chapters should be shown or hidden. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present. This option is the opposite of **chapter/show**. Note that the **scripture/chap/before** and **scripture/chap/after** hooks are still executed when chapters are hidden.

```
\begin{scripture}[Romans 5:1--2][chapter/hide]
  \ch{5}Therefore, since we have been justified
  through faith, we have peace with God through
  our Lord Jesus Christ, \vs{2}through whom we
  have gained access by faith into this grace in
  which we now stand. And we boast in the hope of
  the glory of God.
\end{scripture}
```

[1] Therefore, since we have been justified through faith, we have peace with God through our Lord Jesus Christ, [2] through whom we have gained access by faith into this grace in which we now stand. And we boast in the hope of the glory of God.
Romans 5:1–2

| **chapter/nodrop** | chapter/nodrop = ⟨*boolean*⟩ | Default: false |

Whether the chapter should be set as a drop letter taking up two lines or as a normal letter taking up one line. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present. This option is the opposite of **chapter/drop**.

```
\begin{scripture}[John 1:1--3][chapter/nodrop]
  \ch{1}In the beginning was the Word, and the
  Word was with God, and the Word was God.
  \vs{2}He was with God in the beginning.
  \vs{3}Through him all things were made; without
  him nothing was made that has been made.
\end{scripture}
```

> **1** In the beginning was the Word, and the Word was with God, and the Word was God. ²He was with God in the beginning. ³Through him all things were made; without him nothing was made that has been made.
> **John 1:1–3**

---

chapter/para

chapter/para = ⟨*boolean*⟩   Default: false

New: 2024-01-05   Set chapters in their own paragraph. The spacing before and after the chapter heading can be set using the chapter/para/aboveskip and chapter/para/belowskip options. The value of ⟨boolean⟩ is optional and assumed to be true if not present.

```
\begin{scripture}[John 1:1][chapter/para,
    chapter/format=CHAPTER~#1, version=KJV]
  \ch{1}In the beginning was the Word, and the
  Word was with God, and the Word was God.
\end{scripture}
```

> **CHAPTER 1**
>
> ¹In the beginning was the Word, and the Word was with God, and the Word was God.
> **John 1:1 (KJV)**

---

chapter/para/aboveskip

chapter/para/aboveskip = ⟨*skip*⟩   Default: \bigskipamount

New: 2024-01-05   ⟨skip⟩ added above \ch when chapter/para is in effect. To be consistent with how the parskip package treats headings, no additional \parskip is inserted.

chapter/para/belowskip

chapter/para/belowskip = ⟨*skip*⟩   Default: \medskipamount

New: 2024-01-05   ⟨skip⟩ added below \ch when chapter/para is in effect. To be consistent with how the parskip package treats headings, no additional \parskip is inserted.

chapter/para/indent

chapter/para/indent = ⟨*boolean*⟩   Default: true

New: 2024-01-05   Whether to indent the line following a chapter heading when the chapter/para option is set to true. This only has an effect in the scripture, narrow and midparachap environments. The value of ⟨boolean⟩ is optional and assumed to be true if not present.

chapter/sep

chapter/sep = ⟨*dimen*⟩   Default: 0.5em

⟨dimen⟩ used to separate a \ch from the following text.

chapter/show

chapter/show = ⟨*boolean*⟩   Default: true

New: 2022-08-15   Whether chapters should be shown or hidden. The value of ⟨boolean⟩ is optional and assumed to be true if not present. This option is the opposite of chapter/hide.

chapter/showverse

chapter/showverse = ⟨*boolean*⟩   Default: false

New: 2022-11-14   Whether \vs{1} should be automatically inserted after \ch. The value of ⟨boolean⟩ is optional and assumed to be true if not present. See also midparachap/showverse and midparachap/hideverse. Note that chapter/showverse take precedence over midparachap/hideverse.

```
\begin{scripture}[John 1:1][chapter/showverse]
  \ch{1}In the beginning was the Word, and the
  Word was with God, and the Word was God.
\end{scripture}
```

1 ¹In the beginning was the Word, and the Word was with God, and the Word was God. **John 1:1**

---

chapter/smash = ⟨*boolean*⟩   Default: false

New: 2024-12-30 Whether a chapter should be wrapped in \smash when chapter/nodrop is set. This allows the chapter number to be larger than the line height without affecting line spacing. The value of ⟨boolean⟩ is optional and assumed to be true if not present.

---

chapter/align = bottom | middle | top   Default: bottom

New: 2024-12-30 Sets how a chapter will align vertically when chapter/nodrop is set.

- bottom: Chapter will align at the text base line.

- middle: Chapter will align at the center of text.

- top: Chapter will align at the top of text.

See also chapter/align.

```
\documentclass{article}
\usepackage{babel}
\babelprovide[import, main]{english}
\babelprovide[import,
  onchar=ids fonts]{ancientgreek}
\babelfont[ancientgreek]{rm}[
  BoldFont=NewCM10-Bold]{NewCM10-Regular}
\usepackage{scripture}
\begin{document}
\begin{scripture}[Κατά Μάρκον 1:1--3][
    chapter/font=\huge\bfseries,
    chapter/align=right, chapter/nodrop,
    chapter/smash, chapter/valign=top,
    hanging/hang=1em, hanging/leftmargin=2em,
    hanging/verse/sep=1em, language=ancientgreek,
    verse/font=\sffamily\bfseries,
    verse/sep=0.25em, version=THGNT]
  \begin{hanging}
    \ch{1}Ἀρχὴ τοῦ εὐαγγελίου Ἰησοῦ χριστοῦ υἱοῦ
    θεοῦ· \vs{2}καθὼς γέγραπται ἐν τῷ Ἠσαΐα τῷ
    προφήτη· ἰδοὺ ἀποστέλλω τὸν ἄγγελόν μου πρὸ
    προσώπου σου, ὃς κατασκευάσει τὴν ὁδόν σου.
    \vs{3}φωνὴ βοῶντος ἐν τῇ ἐρήμῳ ἑτοιμάσατε τὴν
    ὁδὸν κυρίου, εὐθείας ποιεῖτε τὰς τρίβους αὐτοῦ.
  \end{hanging}
\end{scripture}
\end{document}
```

1 Ἀρχὴ τοῦ εὐαγγελίου Ἰησοῦ χριστοῦ υἱοῦ θεοῦ· **²** καθὼς γέγραπται ἐν τῷ Ἠσαΐα τῷ προφήτη· ἰδοὺ ἀποστέλλω τὸν ἄγγελόν μου πρὸ προσώπου σου, ὃς κατασκευάσει τὴν ὁδόν σου. **³** φωνὴ βοῶντος ἐν τῇ ἐρήμῳ ἑτοιμάσατε τὴν ὁδὸν κυρίου, εὐθείας ποιεῖτε τὰς τρίβους αὐτοῦ. **Κατά Μάρκον 1:1–3 (THGNT)**

---

chapter/xchar = ⟨*character*⟩   Default: X

New: 2025-01-01 Sets the character to which drop chapters and vertically aligned non-drop chapters are aligned. This can also be set for individual chapters using an optional argument to \ch.

colour
color

New: 2022-08-31

colour = ⟨*colour expression*⟩   Default: .

The l3color ⟨colour expression⟩ to be used for the scripture environment. By default the current colour is inherited. color is an alias for colour.

compact

New: 2022-08-15

compact = ⟨*boolean*⟩   Default: false

This option removes all paragraph breaks along with any \extraskip in the quotation so the whole quotation is a single paragraph of running prose. This option implies chapter/nodrop and all options relating to inner environments are ignored.

The value of ⟨boolean⟩ is optional and assumed to be true if not present.

Warning: This option redefines \par to \relax inside the scripture environment. If you have anything other than expected macros and environments inside your scripture environment TeX will likely get stuck in an infinite loop.

```
\begin{scripture}[Jonah 1:17--2:2a][compact]
  \begin{midparachap}
    \vs{17}Now the \name{Lord} provided a huge
    fish to swallow Jonah, and Jonah was in the
    belly of the fish three days and three
    nights. \ch{2}From inside the fish Jonah
    prayed to the \name{Lord} his God. \vs{2}He
    said:
  \end{midparachap}
  \begin{poetry}
    `In my distress I called to the \name{Lord},
    and he answered me.
  \end{poetry}
\end{scripture}
```

¹⁷ Now the Lᴏʀᴅ provided a huge fish to swallow Jonah, and Jonah was in the belly of the fish three days and three nights. **2** From inside the fish Jonah prayed to the Lᴏʀᴅ his God. ²He said: 'In my distress I called to the Lᴏʀᴅ, and he answered me.

Jonah 1:17–2:2a

defaults

New: 2025-01-01

defaults

Resets options to defaults.

extraskip

extraskip = ⟨*skip*⟩   Default: \medskipamount

⟨skip⟩ used for the \extraskip function.

font

font = ⟨*font*⟩   Default: *Empty*

⟨font⟩ used for the scripture environment.

heading/aboveskip

New: 2024-11-19

heading/aboveskip = ⟨*skip*⟩   Default: \bigskipamount

⟨skip⟩ inserted above a \heading.

heading/afterindent

New: 2024-11-19

heading/afterindent = ⟨*boolean*⟩   Default: false

Whether to indent the first paragraph following a \heading. The value of ⟨boolean⟩ is optional and assumed to be true if not present.

heading/align

New: 2024-11-19

heading/align = left | right | center   Default: left

Set the alignment of a \heading.

heading/belowskip   heading/belowskip = ⟨*skip*⟩   Default: \medskipamount

New: 2024-11-19   ⟨**skip**⟩ inserted below a **\heading**.

heading/font   heading/font = ⟨*font*⟩   Default: \small\itshape

New: 2024-11-19   ⟨**font**⟩ used for a **\heading**.

heading/format   heading/format = ⟨*macro*⟩   Default: #1

New: 2024-11-19   ⟨**macro**⟩ used to format a **\heading**. #1 contains the text of the heading.

heading/hide   heading/hide = ⟨*boolean*⟩   Default: false

New: 2024-11-19   Whether to hide **\heading** functions. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present. This option is the opposite of **heading/show**.

heading/show   heading/show = ⟨*boolean*⟩   Default: true

New: 2024-11-19   Whether to show **\heading** functions. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present. This option is the opposite of **heading/hide**.

indent   indent = ⟨*boolean*⟩   Default: true

Whether to indent the first line of a **scripture** environment or not by **parindent**. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present. This option is the opposite of **noindent**.

inline   inline = ⟨*boolean*⟩   Default: false

New: 2023-01-07   Set the scripture quotation inline. The quotation is surrounded be the values of **inline/begin** and **inline/end** which default to `` `` `` and `` '' ``, respectively. This option also sets the **compact** option and sets the ⟨*reference*⟩ inline. The reference format can be controlled using the following options:

- **inline/reference/format**

- **inline/reference/sep**

- **inline/version/delim**

- **inline/version/format**

Other ⟨*reference*⟩ and ⟨*version*⟩ options are inherited.

     Warning: \par is redefined to \relax when the **inline** option is set. If you have anything other than expected macros and environments inside your quotation TeX will likely get stuck in an infinite loop.

inline/begin   inline/begin = ⟨*text*⟩   Default: `` `` ``

New: 2023-01-07   Text inserted at the beginning of **\textscripture** or a **scripture** environment with the **inline** option set.

inline/end   inline/end = ⟨*text*⟩   Default: `` '' ``

New: 2023-01-07   Text inserted between the end of **\textscripture** or a **scripture** environment with the **inline** option set and the ⟨**reference**⟩ (if specified).

| inline/reference/format | inline/reference/format = ⟨*macro*⟩ | Default: (#1) |

New: 2023-12-27 ⟨macro⟩ used to format the ⟨*reference*⟩ for an inline `scripture` quotation. #1 contains the text of the ⟨*reference*⟩ and ⟨*version*⟩ (if specified).

| inline/reference/sep | inline/reference/sep = ⟨*dimen*⟩ | Default: 0.5em |

New: 2023-12-27 The ⟨dimen⟩ between the end of an inline `scripture` quotation and its ⟨*reference*⟩.

| inline/version/delim | inline/version/delim = ⟨*delim*⟩ | Default: { } |

New: 2023-12-27 ⟨delim⟩ to use between the ⟨*reference*⟩ and the ⟨*version*⟩ of an inline `scripture` quotation.

| inline/version/format | inline/version/format = ⟨*macro*⟩ | Default: #1 |

New: 2023-12-27 ⟨macro⟩ used to format the ⟨*version*⟩ for an inline `scripture` quotation. #1 contains the text of the ⟨*version*⟩.

| language | language = ⟨*language name*⟩ | Default: *Empty* |

New: 2022-11-02 babel or polyglossia ⟨language name⟩ to be used for the `scripture` environment. This calls \selectlanguage{⟨language name⟩} at the start of the environment. Make sure you have babel or polyglossia loaded and correctly set up for your specified language. If unset the current active language is used.

Right-to-left languages are only supported with babel and LuaLATEX. You will need to pass bidi=basic, layout=lists as options to babel to ensure everything is laid out correctly.

```
\documentclass{article}
\usepackage[bidi=basic, layout=lists]{babel}
\babelprovide[import, main]{english}
\babelprovide[import, mapdigits]{persian}
\babelfont[persian]{rm}[
  Renderer=Harfbuzz]{Amiri}
\usepackage{scripture}
\begin{document}
\begin{scripture}[2--1:56 اِشعیا][version=NMV,
    language=persian]
\ch{56}خداوند چنین می‌فرمـاید:
            \begin{poetry}
«انصاف را نگاه داشته،
عدالت را به عمل آورید،

زیـرا نجات من نـزدیک شده،
و عدالت من بـه‌زودی منکشف خـواهد شد.

\vs{2}خوشا بـه حال آن که چنین کند،
و کسی که بدین متمسک گردد؛

که شَبّات را نگاه داشته، آن را بی‌حرمت نسازد،
و دست خویش از هر عمل بد بـاز دارد.»
            \end{poetry}
\end{scripture}
\end{document}
```

۵۶ خداوند چنین می‌فرماید:

«انصاف را نگاه داشته،
عدالت را به عمل آورید،
زیرا نجات من نزدیک شده،
و عدالت من به‌زودی منکشف خواهد شد.
۲خوشا به حال آن که چنین کند،
و کسی که بدین متمسک گردد؛
که شَبّات را نگاه داشته، آن را بی‌حرمت نسازد،
و دست خویش از هر عمل بد باز دارد.»
اِشعیا ۲-۱:۵۶ (NMV)

Note that for CJK scripts the scripture package is compatible with specialist packages like luatexja and ctex. These packages can give nicer kerning and protrusion out of the box. Although with some effort you can set up babel along with microtype to give comparable output.

```
\documentclass{article}
\usepackage{luatexja}
\usepackage{scripture}
\begin{document}
\begin{scripture}[マルコの福音書 9:33--37][
    version=NJB]
  \vs{33}一行はカペナウムに着いた。イエスは家に入ってか
      ら、弟子たちにお尋ねになった。「来る途中、何を
      論じ合っていたのですか。」
  \vs{34}彼らは黙っていた。来る途中、だれが一番偉いか論
      じ合っていたからである。
  \vs{35}イエスは腰を下ろすと、十二人を呼んで言われた。
      「だれでも先頭に立ちたいと思う者は、皆の後にな
      り、皆に仕える者になりなさい。」
  \vs{36}それから、イエスは一人の子どもの手を取って、彼
      らの真ん中に立たせ、腕に抱いて彼らに言われた。
  \vs{37}「だれでも、このような子どもたちの一人を、わた
      しの名のゆえに受け入れる人は、わたしを受け入れ
      るのです。また、だれでもわたしを受け入れる人
      は、わたしではなく、わたしを遣わされた方を受け入
      れるのです。」
\end{scripture}
\end{document}
```

³³一行はカペナウムに着いた。イエスは家に入ってから、弟子たちにお尋ねになった。「来る途中、何を論じ合っていたのですか。」³⁴彼らは黙っていた。来る途中、だれが一番偉いか論じ合っていたからである。³⁵イエスは腰を下ろすと、十二人を呼んで言われた。「だれでも先頭に立ちたいと思う者は、皆の後になり、皆に仕える者になりなさい。」³⁶それから、イエスは一人の子どもの手を取って、彼らの真ん中に立たせ、腕に抱いて彼らに言われた。³⁷「だれでも、このような子どもたちの一人を、わたしの名のゆえに受け入れる人は、わたしを受け入れるのです。また、だれでもわたしを受け入れる人は、わたしではなく、わたしを遣わされた方を受け入れるのです。」

マルコの福音書 9:33–37 (NJB)

---

language/variant

New: 2022-11-02

language/variant = ⟨*language variant name*⟩   Default: *Empty*

polyglossia ⟨language variant name⟩ to be used along with the specified language for the scripture environment. This inserts `\selectlanguage[`⟨*language variant name*⟩`]{`⟨*language name*⟩`}` at the start of the environment. Make sure you have polyglossia loaded and correctly set up for your specified language.

---

leftmargin

leftmargin = ⟨*dimen*⟩   Default: 0pt

⟨dimen⟩ used for the left margin of the scripture environment.

---

name/font

New: 2022-09-01

name/font = ⟨*font*⟩   Default: \scshape

⟨font⟩ used for the \name function.

---

name/format

New: 2022-09-01

name/format = ⟨*macro*⟩   Default: #1

⟨macro⟩ used to format the divine name using the \name function. #1 contains the text of the argument to \name.

---

noindent

noindent = ⟨*boolean*⟩   Default: false

Whether to indent the first line of a scripture environment or not by parindent. The value of ⟨boolean⟩ is optional and assumed to be true if not present. This option is the opposite of indent.

```
\begin{scripture}[1 Peter 3:18][noindent]
  \vs{18}For Christ also suffered once for sins,
  the righteous for the unrighteous, to bring you
  to God. He was put to death in the body but
  made alive in the Spirit.
\end{scripture}
```

> [18]For Christ also suffered once for sins, the righteous for the unrighteous, to bring you to God. He was put to death in the body but made alive in the Spirit. **1 Peter 3:18**

---

parindent　　parindent = ⟨*dimen*⟩　　　　　　　　　　　　　　　　Default: \parindent

⟨dimen⟩ to use when indenting the first line of a paragraph in the **scripture** environment.

---

parskip　　parskip = ⟨*skip*⟩　　　　　　　　　　　　　　　　　　Default: \parskip

⟨skip⟩ to use between paragraphs in the **scripture** environment.

---

pilcrow　　pilcrow = ⟨*boolean*⟩　　　　　　　　　　　　　　　　　Default: false

New: 2024-01-18　Whether to insert a pilcrow (¶) at the beginning of each paragraph. This option only has an effect when using the **compact** and/or **verse/para** options. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present.

A subsequent pilcrow can be suppressed with the **\nopilcrow** macro. Inner environments start a new paragraph only if there is a blank line before them and similarly, there is only a new paragraph after an inner environment if it is followed by a blank line. New paragraphs are not inserted within a poetry environment, so to obtain a paragraph break within a block of poetry, use two **poetry** environments separated by a blank line. The exception to this is the **midparachap** environment which always starts and ends a paragraph.

```
\begin{scripture}[Deuteronomy 5:1][chapter/para,
    chapter/format=CHAPTER~#1, verse/para,
    verse/font=\bfseries, verse/format=#1,
    verse/sep=0.5em, pilcrow]
  \ch{5}Moses summoned all Israel and said:

  Hear, Israel, the decrees and the laws I declare
  in your hearing today. Learn them and be sure to
  follow them.
\end{scripture}
```

> **CHAPTER 5**
>
> **1** ¶ Moses summoned all Israel and said: ¶ Hear, Israel, the decrees and the laws I declare in your hearing today. Learn them and be sure to follow them. **Deuteronomy 5:1**

---

pilcrow/sep　　pilcrow/sep = ⟨*dimen*⟩　　　　　　　　　　　　　　　Default: 0.25em

New: 2024-01-18　⟨dimen⟩ used to separate a pilcrow (¶) from the following text.

---

redletter　　redletter = ⟨*boolean*⟩　　　　　　　　　　　　　　　Default: false

New: 2022-08-31　Whether to enable highlighting of Jesus' words in red. The value of ⟨boolean⟩ is optional and assumed to be **true** if not present. The text needs to be marked up with **\redletteron** and **\redletteroff** for this option to have any effect.

redletter/colour	redletter/colour = ⟨*colour expression*⟩	Default: red!80!black
redletter/color		

New: 2022-08-31 The l3color ⟨`colour expression`⟩ to be used for the words of Jesus when the `redletter` option is true. `\redletteron` and `\redletteroff` must be used to mark up the text. `redletter/color` is an alias for `redletter/colour`.

reference/align	reference/align = left | right	Default: right

New: 2023-01-08 Set the alignment of the reference. Note that in neither case will the reference break across lines.

reference/colour	reference/colour = ⟨*colour expression*⟩	Default: .
reference/color		

New: 2022-09-07 The l3color ⟨`colour expression`⟩ to be used for the reference and version. By default the current colour is inherited. `reference/color` is an alias for `reference/colour`.

reference/font	reference/font = ⟨*font*⟩	Default: \bfseries

⟨*font*⟩ used when printing the ⟨*reference*⟩ for a `scripture` quotation.

reference/format	reference/format = ⟨*macro*⟩	Default: #1

⟨*macro*⟩ used to format the ⟨*reference*⟩ for a `scripture` quotation. `#1` contains the text of the ⟨*reference*⟩ and ⟨*version*⟩ (if specified).

reference/newline	reference/newline

New: 2023-01-8 Always push the ⟨*reference*⟩ to a newline even if there is space for it to fit on the last line of the quotation. The option sets `reference/sep` to `\linewidth` so there is never room for the reference on the final line of the quotation.

reference/position	reference/position = start | end	Default: end

New: 2024-01-10 Set the position of the reference at either the start or end of the quotation. If `start` is specified, then the reference will be set inline unless the quotation begins with an inner environment. Any `\vs` or `\ch` is suppressed at the start of the quotation in the main `scripture` environment.

```
\begin{scripture}[John 1:1][version=NIV,
    reference/position=start]
  \ch{1}In the beginning was the Word, and the
  Word was with God, and the Word was God.
\end{scripture}
```

> **John 1:1 (NIV)** In the beginning was the Word, and the Word was with God, and the Word was God.

reference/sep	reference/sep = ⟨*dimen*⟩	Default: 2em

The minimum ⟨*dimen*⟩ allowed between the end of a `scripture` quotation and its ⟨*reference*⟩ before the ⟨*reference*⟩ is pushed to the following line.

reference/start/inline reference/start/inline = ⟨*boolean*⟩   Default: true

New: 2024-01-10   When `reference/position` is set to `start`, this option controls whether the reference is set inline or on a line of its own before the quotation. Note that if the quotation begins with an inner environment the reference will always be on a new line unless the `compact` option is also set. The value of ⟨boolean⟩ is optional and assumed to be `true` if not present. This option is the opposite of `reference/start/newline`.

reference/start/newline reference/start/newline = ⟨*boolean*⟩   Default: false

New: 2024-01-10   When `reference/position` is set to `start`, this option controls whether the reference is set inline or on a line of its own before the quotation. Note that if the quotation begins with an inner environment the reference will always be on a new line unless the `compact` option is also set. The value of ⟨boolean⟩ is optional and assumed to be `true` if not present. This option is the opposite of `reference/start/inline`.

```
\begin{scripture}[John 1:1]%
    [reference/position=start,
    reference/start/newline]
  \ch{1}In the beginning was the Word, and the
  Word was with God, and the Word was God.
\end{scripture}
```

John 1:1
1 In the beginning was the Word, and the Word was with God, and the Word was God.

reference/start/sep reference/start/sep = ⟨*dimen*⟩   Default: 1em

New: 2024-01-10   The space inserted between the reference and the quotation when `reference/position` is set to `start`. This only applies when the reference is inline and not on a line of its own.

reference/start/vsep reference/start/vsep = ⟨*skip*⟩   Default: 0pt

New: 2024-01-10   The vertical skip inserted between the reference and the quotation when `reference/position` is set to `start`. This only applies when the reference is on a line of its own.

rightmargin rightmargin = ⟨*dimen*⟩   Default: 0pt

⟨dimen⟩ used for the right margin of the `scripture` environment.

selah/font selah/font = ⟨*font*⟩   Default: \itshape

⟨font⟩ used when printing `\selah` in a `poetry` environment.

selah/format selah/format = ⟨*macro*⟩   Default: #1

⟨macro⟩ used to format the `\selah` in a `poetry` environment. `#1` contains the value of `selah/text`.

selah/sep selah/sep = ⟨*dimen*⟩   Default: 1em

The minimum ⟨dimen⟩ allowed between the end of a poetry line and *Selah* before the *Selah* is pushed to the following line.

`selah/text`	`selah/text = ⟨text⟩`	Default: Selah

⟨text⟩ used to for **\selah** in a **poetry** environment.

`style`	`style = ⟨name⟩`	
New: 2023-12-27		

Set options previously saved with **\scripturestyle**. An error occurs if the style isn't defined.

`textright/sep`	`textright/sep = ⟨dimen⟩`	Default: 1em
New: 2022-08-30		

The minimum ⟨dimen⟩ allowed between the end of a line and text aligned to the right with **\textright** before the right aligned text is pushed to the following line.

`verse/colour`	`verse/colour = ⟨colour expression⟩`	Default: .
`verse/color`		
New: 2022-09-07		

The l3color ⟨colour expression⟩ to be used for verse numbers. By default the current colour is inherited. **verse/color** is an alias for **verse/colour**.

`verse/first`	`verse/first = ⟨boolean⟩`	Default: false
New: 2022-11-13		

In prose sections, print the verse number in a different format if it is at the start of the paragraph. The format and separation between verse and text are controlled with the **verse/firstformat** and **verse/firstsep** options. Use **\nofirstverse** to suppress this at the next verse (e.g., if your quote starts mid-way though a paragraph and a normal verse should be printed). The value of ⟨boolean⟩ is optional and assumed to be true if not present.

```
\begin{scripture}[Luke 18:25--28][noindent,
    verse/first, version=NRSV]
  \nofirstverse\vs{25}Indeed, it is easier for a
  camel to go through the eye of a needle than for
  someone who is rich to enter the kingdom of
  God.''

  \vs{26}Those who heard it said, ``Then who can
  be saved?'' \vs{27}He replied, ``What is
  impossible for mortals is possible for God.''

  \vs{28}Then Peter said, ``Look, we have left our
  homes and followed you.''
\end{scripture}
```

^{25}Indeed, it is easier for a camel to go through the eye of a needle than for someone who is rich to enter the kingdom of God."

26 Those who heard it said, "Then who can be saved?" ^{27}He replied, "What is impossible for mortals is possible for God."

28 Then Peter said, "Look, we have left our homes and followed you."
Luke 18:25–28 (NRSV)

`verse/firstformat`	`verse/firstformat = ⟨macro⟩`	Default: #1
New: 2022-11-13		

⟨macro⟩ used to format the first **\vs** in a paragraph. #1 contains the argument of **\vs**. Only used if **verse/first** is true.

`verse/firstsep`	`verse/firstsep = ⟨dimen⟩`	Default: 0.5em
New: 2022-11-13		

⟨dimen⟩ used to separate the first **\vs** in a paragraph from the following text. Only used if **verse/first** is true.

verse/font	verse/font = ⟨*font*⟩	Default: *Empty*

⟨`font`⟩ used for the `\vs` function.

verse/format	verse/format = ⟨*macro*⟩	Default: ^{#1}

⟨`macro`⟩ used to format the `\vs` function. `#1` contains the argument of `\vs`.

verse/hide	verse/hide = ⟨*boolean*⟩	Default: false

Whether verses should be shown or hidden. The value of ⟨`boolean`⟩ is optional and assumed to be `true` if not present. This option is the opposite of verse/show. Note that the scripture/verse/before and scripture/verse/after hooks are still executed when verses are hidden.

```
\begin{scripture}[Colossians 1:15--16][verse/hide]
  \vs{15}The Son is the image of the invisible
  God, the firstborn over all creation.
  \vs{16}For in him all things were created:
  things in heaven and on earth, visible and
  invisible, whether thrones or powers or rulers
  or authorities; all things have been created
  through him and for him.
\end{scripture}
```

> The Son is the image of the invisible God, the firstborn over all creation. For in him all things were created: things in heaven and on earth, visible and invisible, whether thrones or powers or rulers or authorities; all things have been created through him and for him.
> **Colossians 1:15–16**

verse/para	verse/para = ⟨*boolean*⟩	Default: false

Whether to set each verse on its own line. The value of ⟨`boolean`⟩ is optional and assumed to be `true` if not present. Drop chapters and the shape of inner environments is preserved with this option unless the compact option is also set.

```
\begin{scripture}[Jonah 1:17--2:2a][verse/para,
    verse/sep=0.5em, pilcrow, chapter/para,
    chapter/format=CHAPTER~#1]
  \begin{midparachap}
    \vs{17}Now the \name{Lord} provided a huge
    fish to swallow Jonah, and Jonah was in the
    belly of the fish three days and three
    nights. \ch{2}From inside the fish Jonah
    prayed to the \name{Lord} his God. \vs{2}He
    said:
  \end{midparachap}

  \begin{poetry}
    `In my distress I called to the \name{Lord},
    and he answered me.
  \end{poetry}
\end{scripture}
```

> **17** ¶ Now the LORD provided a huge fish to swallow Jonah, and Jonah was in the belly of the fish three days and three nights.
>
> **CHAPTER 2**
>
> **1** From inside the fish Jonah prayed to the LORD his God. **2** He said:
>
> ¶ 'In my distress I called to the LORD,
>   and he answered me.
> **Jonah 1:17–2:2a**

```
\begin{scripture}[Jonah 1:17--2:2a][verse/para,
    verse/sep=0.5em, pilcrow, chapter/para,
    chapter/format=CHAPTER~#1, compact]
  \begin{midparachap}
    \vs{17}Now the \name{Lord} provided a huge
    fish to swallow Jonah, and Jonah was in the
    belly of the fish three days and three
    nights. \ch{2}From inside the fish Jonah
    prayed to the \name{Lord} his God. \vs{2}He
    said:
  \end{midparachap}

  \begin{poetry}
    `In my distress I called to the \name{Lord},
    and he answered me.
  \end{poetry}
\end{scripture}
```

17 ¶ Now the LORD provided a huge fish to swallow Jonah, and Jonah was in the belly of the fish three days and three nights.

CHAPTER 2

1 From inside the fish Jonah prayed to the LORD his God. **2** He said: ¶ 'In my distress I called to the LORD, and he answered me.

Jonah 1:17–2:2a

---

verse/sep   verse/sep = ⟨*dimen*⟩   Default: 0.05em

New: 2022-07-28   ⟨dimen⟩ used to separate a **\vs** from the following text.

---

verse/show   verse/show = ⟨*boolean*⟩   Default: true

New: 2022-08-15   Whether verses should be shown or hidden. The value of ⟨boolean⟩ is optional and assumed to be `true` if not present. This option is the opposite of **verse/hide**.

---

version   version = ⟨*version*⟩   Default: *Empty*

⟨*version*⟩ added to the ⟨*reference*⟩ for a scripture quotation.

---

version/delim version/delim = ⟨*delim*⟩   Default: { }

⟨delim⟩ to use between the ⟨*reference*⟩ and the ⟨*version*⟩ of a scripture quotation.

---

version/format version/format = ⟨*macro*⟩   Default: (#1)

⟨macro⟩ used to format the ⟨*version*⟩ for a scripture quotation. #1 contains the text of the ⟨*version*⟩.

---

version/position version/position = withref | end   Default: withref

New: 2024-01-10   Set the position of the version at either with the reference or at the end of the quotation if **reference/position** is set to `start`.

```
\begin{scripture}[John 1:1][version=NIV,
    reference/position=start,
    version/position=end]
  \ch{1}In the beginning was the Word, and the
  Word was with God, and the Word was God.
\end{scripture}
```

John 1:1 In the beginning was the Word, and the Word was with God, and the Word was God.

(NIV)

2.4.1 Options for the `center` environment

If these options are passed directly to the `center` environment rather than to the package, `\scripturesetup` or the `scripture` environment, then the `center/` prefix should be dropped.

center/aboveskip	center/aboveskip = ⟨*skip*⟩	Default: \medskipamount

⟨`skip`⟩ inserted above the `center` environment.

center/belowskip	center/belowskip = ⟨*skip*⟩	Default: \medskipamount

⟨`skip`⟩ inserted below the `center` environment.

center/leftmargin	center/leftmargin = ⟨*dimen*⟩	Default: 0pt

⟨`dimen`⟩ used for the left margin of the `center` environment.

center/rightmargin	center/rightmargin = ⟨*dimen*⟩	Default: 0pt

⟨`dimen`⟩ used for the right margin of the `center` environment.

2.4.2 Options for the `flushright` environment

If these options are passed directly to the `flushright` environment rather than to the package, `\scripturesetup` or the `scripture` environment, then the `flushright/` prefix should be dropped.

flushright/aboveskip	flushright/aboveskip = ⟨*skip*⟩	Default: \medskipamount

⟨`skip`⟩ inserted above the `flushright` environment.

flushright/belowskip	flushright/belowskip = ⟨*skip*⟩	Default: \medskipamount

⟨`skip`⟩ inserted below the `flushright` environment.

flushright/leftmargin	flushright/leftmargin = ⟨*dimen*⟩	Default: 0pt

⟨`dimen`⟩ used for the left margin of the `flushright` environment.

flushright/rightmargin	flushright/rightmargin = ⟨*dimen*⟩	Default: 0pt

⟨`dimen`⟩ used for the right margin of the `flushright` environment.

2.4.3 Options for the `hanging` environment

If these options are passed directly to the `hanging` environment rather than to the package, `\scripturesetup` or the `scripture` environment, then the `hanging/` prefix should be dropped.

hanging/aboveskip	hanging/aboveskip = ⟨*skip*⟩	Default: \medskipamount

⟨`skip`⟩ inserted above the `hanging` environment.

hanging/belowskip = ⟨*skip*⟩   Default: \medskipamount

⟨skip⟩ inserted below the hanging environment.

hanging/chapter/indent = ⟨*boolean*⟩   Default: true

New: 2024-12-31 Indent the first line of chapters in hanging environments. If this is set to false, then the paragraph will be set has a normal hanging paragraph (if there is room).

```
\begin{scripture}[Genesis 1:1--3][%
    hanging/chapter/indent=false]
  \begin{hanging}
    \ch{1}In the beginning God created the heavens
    and the earth. \vs{2}Now the earth was
    formless and empty, darkness was over the
    surface of the deep, and the Spirit of God was
    hovering over the waters.

    \vs{3}And God said, ``Let there be light,''
    and there was light.
  \end{hanging}
\end{scripture}
```

1 In the beginning God created the heavens and the earth. ²Now the earth was formless and empty, darkness was over the surface of the deep, and the Spirit of God was hovering over the waters.
³And God said, "Let there be light," and there was light.
                                                   Genesis 1:1–3

hanging/hang = ⟨*dimen*⟩   Default: 2em

⟨dimen⟩ used for the indent of subsequent lines of the hanging environment.

hanging/leftmargin = ⟨*dimen*⟩   Default: 1em

⟨dimen⟩ used for the left margin of the hanging environment.

hanging/rightmargin = ⟨*dimen*⟩   Default: 1em

⟨dimen⟩ used for the right margin of the hanging environment.

hanging/verse/left = ⟨*boolean*⟩   Default: false

New: 2024-12-17 Align verse numbers in hanging environments at the left margin of the hanging environment. This is only used for verses at the start of a paragraph. This option is the opposite of hanging/verse/right.

```
\documentclass{article}
\usepackage{babel}
\babelprovide[import, main]{english}
\babelprovide[import,
  onchar=ids fonts]{ancientgreek}
\babelfont[ancientgreek]{rm}[
  BoldFont=NewCM10-Bold]{NewCM10-Regular}
\usepackage{scripture}
\begin{document}
\begin{scripture}[Κατά Μάρκον 1:28--30][
    hanging/hang=1em, hanging/leftmargin=2em,
    hanging/verse/left, hanging/verse/sep=1em,
    language=ancientgreek, verse/sep=0.25em,
    verse/font=\sffamily\bfseries, version=THGNT]
  \begin{hanging}
    \vs{28}Καὶ ἐξῆλθεν ἡ ἀκοὴ αὐτοῦ εὐθὺς πανταχοῦ
    εἰς ὅλην τὴν περίχωρον τῆς Γαλιλαίας.

    \vs{29}Καὶ εὐθὺς ἐκ τῆς συναγωγῆς ἐξελθόντες
    ἦλθον εἰς τὴν οἰκίαν Σίμωνος καὶ Ἀνδρέου μετὰ
    Ἰακώβου καὶ Ἰωάννου. \vs{30}ἡ δὲ πενθερὰ
    Σίμωνος κατέκειτο πυρέσσουσα, καὶ εὐθὺς
    λέγουσιν αὐτῷ περὶ αὐτῆς.
  \end{hanging}
\end{scripture}
\end{document}
```

28 Καὶ ἐξῆλθεν ἡ ἀκοὴ αὐτοῦ εὐθὺς πανταχοῦ εἰς ὅλην τὴν περίχωρον τῆς Γαλιλαίας.
29 Καὶ εὐθὺς ἐκ τῆς συναγωγῆς ἐξελθόντες ἦλθον εἰς τὴν οἰκίαν Σίμωνος καὶ Ἀνδρέου μετὰ Ἰακώβου καὶ Ἰωάννου. **30** ἡ δὲ πενθερὰ Σίμωνος κατέκειτο πυρέσσουσα, καὶ εὐθὺς λέγουσιν αὐτῷ περὶ αὐτῆς.

Κατά Μάρκον 1:28–30 (THGNT)

---

hanging/verse/right　　hanging/verse/right = ⟨*boolean*⟩　　　　　　　　　　　　Default: true

New: 2024-12-17　Align verse numbers in `hanging` environments at the left margin of the `hanging` environment. The verse numbers overlap to the left. This is only used for verses at the start of a paragraph. This option is the opposite of `hanging/verse/left`.

---

hanging/verse/sep　　hanging/verse/sep = ⟨*dimen*⟩　　　　　　　　　　　　Default: 0.05em

New: 2024-12-17　⟨*dimen*⟩ used to separate a `\vs` from the following text in a `hanging` environment. This is only used for verses at the start of a paragraph.

2.4.4　Options for the `midparachap` environment

If these options are passed directly to the `midparachap` environment rather than to the package, `\scripturesetup` or the `scripture` environment, then the `midparachap/` prefix should be dropped.

---

midparachap/hideverse　　midparachap/hideverse = ⟨*boolean*⟩　　　　　　　　Default: false

Whether a `\vs{1}` should automatically be inserted after a `\ch` when the `\ch` occurs mid-paragraph. This also affects `\ch` in `poetry` environments. The value of ⟨*boolean*⟩ is optional and assumed to be `true` if not present. This option is the opposite of `midparachap/showverse`. See also `chapter/showverse`.

```
\begin{scripture}[Hosea 11:12--12:1]%
    [midparachap/hideverse]
  \begin{poetry}
    \vs{12}Ephraim has surrounded me with lies,
    Israel with deceit.

    And Judah is unruly against God,
    even against the faithful Holy One.

    \ch{12}Ephraim feeds on the wind;
    he pursues the east wind all day
    and multiplies lies and violence.

    He makes a treaty with Assyria
    and sends olive oil to Egypt.
  \end{poetry}
\end{scripture}
```

[12]Ephraim has surrounded me with
lies,
Israel with deceit.
And Judah is unruly against God,
even against the faithful Holy
One.
12 Ephraim feeds on the wind;
he pursues the east wind all
day
and multiplies lies and violence.
He makes a treaty with Assyria
and sends olive oil to Egypt.
Hosea 11:12–12:1

---

midparachap/showverse   midparachap/showverse = ⟨*boolean*⟩   Default: true

Whether a `\vs{1}` should automatically be inserted after a `\ch` when the `\ch` occurs mid-paragraph. This also affects `\ch` in `poetry` environments. The value of ⟨boolean⟩ is optional and assumed to be `true` if not present. This option is the opposite of `midparachap/hideverse`. See also `chapter/showverse`.

2.4.5   Options for the `narrow` environment

If these options are passed directly to the `narrow` environment rather than to the package, `\scripturesetup` or the `scripture` environment, then the `narrow/` prefix should be dropped.

narrow/aboveskip   narrow/aboveskip = ⟨*skip*⟩   Default: \medskipamount

⟨skip⟩ inserted above the `narrow` environment.

narrow/belowskip   narrow/belowskip = ⟨*skip*⟩   Default: \medskipamount

⟨skip⟩ inserted below the `narrow` environment.

narrow/leftmargin   narrow/leftmargin = ⟨*dimen*⟩   Default: 1em

⟨dimen⟩ used for the left margin of the `narrow` environment.

narrow/rightmargin   narrow/rightmargin = ⟨*dimen*⟩   Default: 1em

⟨dimen⟩ used for the right margin of the `narrow` environment.

2.4.6   Options for the `poetry` environment

If these options are passed directly to the `poetry` environment rather than to the package, `\scripturesetup` or the `scripture` environment, then the `poetry/` prefix should be dropped.

poetry/aboveskip | poetry/aboveskip = ⟨*skip*⟩ | Default: \medskipamount

⟨skip⟩ inserted above the poetry environment.

poetry/belowskip | poetry/belowskip = ⟨*skip*⟩ | Default: \medskipamount

⟨skip⟩ inserted below the poetry environment.

poetry/bigindent | poetry/bigindent = ⟨*dimen*⟩ | Default: 4em

⟨dimen⟩ used for the indent subsequent lines of a line of poetry in a poetry environment if it needs to flow on to the following line.

poetry/indent | poetry/indent = ⟨*dimen*⟩ | Default: 1em

⟨dimen⟩ used for the second half of a line of poetry in a poetry environment.

poetry/leftmargin | poetry/leftmargin = ⟨*dimen*⟩ | Default: 1em

⟨dimen⟩ used for the left margin of the poetry environment.

poetry/rightmargin | poetry/rightmargin = ⟨*dimen*⟩ | Default: 0pt

⟨dimen⟩ used for the right margin of the poetry environment.

poetry/verse/left | poetry/verse/left = ⟨*boolean*⟩ | Default: false

New: 2022-07-31 — Align verse numbers in poetry environments at the left margin of the scripture environment. This option is the opposite of poetry/verse/right.

```
\begin{scripture}[Acts 15:15--18]%
    [poetry/leftmargin=2em, poetry/verse/left]
  \vs{15}The words of the prophets are in
  agreement with this, as it is written:
  \begin{poetry}
    \vs{16}`\thinspace``After this I will return
    and rebuild David's fallen tent.

    Its ruins I will rebuild,
    and I will restore it,

    \vs{17}that the rest of mankind may seek the Lord,
    even all the Gentiles who bear my name,

    says the Lord, who does these things''---
    \vs{18}things known from long ago.
  \end{poetry}
\end{scripture}
```

> ¹⁵The words of the prophets are in agreement with this, as it is written:
>
> ¹⁶ ' "After this I will return
>   and rebuild David's fallen tent.
>   Its ruins I will rebuild,
>   and I will restore it,
> ¹⁷ that the rest of mankind
>   may seek the Lord,
>   even all the Gentiles who
>   bear my name,
>   says the Lord, who does
>   these things"—
> ¹⁸ things known from long ago.
>   **Acts 15:15–18**

poetry/verse/right | poetry/verse/right = ⟨*boolean*⟩ | Default: true

New: 2022-07-31 — Align verse numbers in poetry environments at the left margin of the poetry environment. The verse numbers overlap to the left. This option is the opposite of poetry/verse/left.

poetry/verse/sep | poetry/verse/sep = ⟨*dimen*⟩ | Default: 0.05em

New: 2022-07-31 — ⟨dimen⟩ used to separate a \vs from the following text in a poetry environment. This is only used for verses at the start of a line.

30

3 Implementation

```
1 %<*package>
2 %<@@=scripture>

3 \NeedsTeXFormat{LaTeX2e}[2022-11-01]
4 \ProvidesExplPackage{scripture}{2025/01/02}{2.1}
5   {Format Scripture Quotations (DCP)}
```

3.1 Hacks and workarounds

Work around odd interaction between `\textdir` and display math. See https://tex.stackexchange.com/q/716621/87678.

```
6 \bool_new:N \l__scripture_textdir_change_in_group_bool
7 \cs_new_protected:Nn \__scripture_set_textdirection:
8   {
9     \sys_if_engine_luatex:T
10       {
11         \textdirection \int_use:N \textdirection
12         \bool_set_true:N \l__scripture_textdir_change_in_group_bool
13       }
14   }
```

3.2 Messages

`scripture` environments along with all internal environments can not be nested.

```
15 \msg_new:nnnn { scripture } { nested-environment }
16   { Nesting~of~scripture~environments~not~supported. }
17   { You~have~tried~to~nest~a~scripture~environment~or \\
18     one~of~its~internal~environments,~which~is~not~supported. }
```

Make sure polyglossia languages and variants are set up.

```
19 \msg_new:nnnn { scripture } { language-error }
20   { Language~`#1'~not~set~up. }
21   { Language~`#1'~has~not~been~set~up.~Make~sure~you~set~up~languages~
22     before~using~the~language~option~of~the~scripture~environment. }
```

Warn if a `language` is requested and neither babel nor polyglossia is loaded.

```
23 \msg_new:nnnn { scripture } { language-warning }
24   { Language~requested~without~support. }
25   { You~have~requested~language=#1,~but~neither~babel~nor~polyglossia~
26     have~been~loaded~and~set~up.~Ignoring~option~language=#1. }
```

Error if an `\heading` is not used in the main `scripture` environment.

```
27 \msg_new:nnnn { scripture } { heading-error }
28   { \token_to_str:N \heading \c_space_tl not~support~within~inner~
29     environments. }
30   { You~have~tried~to~use~\token_to_str:N \heading \c_space_tl within~the~
31     `#1'~environment,~but~this~macro~is~only~supported~in~the~main~
32     `scripture'~environment. }
```

Error if an unknown `style` is requested.

```
33 \msg_new:nnnn { scripture } { unknown-style }
34   { Style~#1~not~defined. }
35   { You~have~requested~style=#1,~but~it~has~not~been~defined.~
36     You~can~define~it~using~\token_to_str:N \scripturestyle.~
37     Ignoring~option~style=#1. }
```

3.3 Options

Key-value settings for the `scripture` environment and its inner environments.

```
38 \keys_define:nn { scripture }
39   {
40     , aboveskip              .tl_set:N          = \l__scripture_aboveskip_tl
41     , aboveskip              .value_required:n  = true
42     , aboveskip              .initial:n         = \c_zero_skip
43     , added / font           .tl_set:N          = \l__scripture_added_font_tl
44     , added / font           .value_required:n  = true
45     , added / font           .initial:n         =
46     , added / format         .cs_set:Np         = \__scripture_added_format:n #1
47     , added / format         .value_required:n  = true
48     , added / format         .initial:n         = \emph { #1 }
49     , after                  .tl_set:N          = \l__scripture_after_tl
50     , after                  .value_required:n  = true
51     , after                  .initial:n         =
52     , before                 .tl_set:N          = \l__scripture_before_tl
53     , before                 .value_required:n  = true
54     , before                 .initial:n         =
55     , belowskip              .tl_set:N          = \l__scripture_belowskip_tl
56     , belowskip              .value_required:n  = true
57     , belowskip              .initial:n         = \c_zero_skip
58     , colour                 .tl_set:N          = \l__scripture_colour_tl
59     , colour                 .value_required:n  = true
60     , colour                 .initial:n         = .
61     , color                  .meta:n            = { colour = { #1 } }
62     , compact                .bool_set:N        = \l__scripture_compact_bool
63     , compact                .default:n         = true
64     , compact                .initial:n         = false
65     , chapter / align        .choices:nn        = { left, right }
66                                                  {
67                                                    \tl_if_exist:NF \l__scripture_chap
68                                                      { \tl_new:N \l__scripture_chapte
69                                                    \str_case:Vn \l_keys_choice_tl
70                                                      {
71                                                        { left }
72                                                          {
73                                                            \tl_set:Nn \l__scripture_c
74                                                              { l }
75                                                          }
76                                                        { right }
77                                                          {
78                                                            \tl_set:Nn \l__scripture_c
79                                                              { r }
80                                                          }
```

```
81                                                                      }
82                                                                    }
83      , chapter / align            .value_required:n   = true
84      , chapter / align            .initial:n          = left
85      , chapter / font             .tl_set:N           = \l__scripture_chap_font_tl
86      , chapter / font             .value_required:n   = true
87      , chapter / font             .initial:n          = \bfseries
88      , chapter / colour           .tl_set:N           = \l__scripture_chapter_colour_tl
89      , chapter / colour           .value_required:n   = true
90      , chapter / colour           .initial:n          = .
91      , chapter / color            .meta:n             = { chapter / colour = { #1 } }
92      , chapter / drop             .bool_set:N         = \l__scripture_chap_drop_bool
93      , chapter / drop             .default:n          = true
94      , chapter / drop             .initial:n          = true
95      , chapter / format           .cs_set:Np          = \__scripture_chap_format:n #1
96      , chapter / format           .value_required:n   = true
97      , chapter / format           .initial:n          = #1
98      , chapter / hide             .bool_set_inverse:N = \l__scripture_chap_show_bool
99      , chapter / hide             .default:n          = true
100     , chapter / nodrop           .bool_set_inverse:N = \l__scripture_chap_drop_bool
101     , chapter / nodrop           .default:n          = true
102     , chapter / para             .bool_set:N         = \l__scripture_chap_para_bool
103     , chapter / para             .default:n          = true
104     , chapter / para             .initial:n          = false
105     , chapter / para / aboveskip .tl_set:N           = \l__scripture_chap_para_aboveskip_tl
106     , chapter / para / aboveskip .value_required:n   = true
107     , chapter / para / aboveskip .initial:n          = \bigskipamount
108     , chapter / para / belowskip .tl_set:N           = \l__scripture_chap_para_belowskip_tl
109     , chapter / para / belowskip .value_required:n   = true
110     , chapter / para / belowskip .initial:n          = \medskipamount
111     , chapter / para / indent    .code:n             = {
112                                                            \bool_set:Nn
113                                                              \l__scripture_chap_para_afterind
114                                                              { \str_if_eq_p:nn { #1 } { true
115                                                          }
116     , chapter / para / indent    .default:n          = true
117     , chapter / para / indent    .initial:n          = true
118     , chapter / sep              .tl_set:N           = \l__scripture_chap_sep_tl
119     , chapter / sep              .value_required:n   = true
120     , chapter / sep              .initial:n          = 0.5em
121     , chapter / show             .bool_set:N         = \l__scripture_chap_show_bool
122     , chapter / show             .default:n          = true
123     , chapter / show             .initial:n          = true
124     , chapter / showverse        .bool_set:N         = \l__scripture_chap_show_verse_bool
125     , chapter / showverse        .default:n          = true
126     , chapter / showverse        .initial:n          = false
127     , chapter / smash            .bool_set:N         = \l__scripture_chap_smash_bool
128     , chapter / smash            .default:n          = true
129     , chapter / smash            .initial:n          = false
130     , chapter / valign           .choices:nn         = { bottom, middle, top }
131                                                        {
132                                                          \tl_if_exist:NF \l__scripture_chap
133                                                            { \tl_new:N \l__scripture_chapte
134                                                          \str_case:Vn \l_keys_choice_tl
```

```
135                                                                          {
136                                                                            { bottom }
137                                                                              {
138                                                                                \tl_set:Nn \l__scripture_c
139                                                                                  { b }
140                                                                              }
141                                                                            { middle }
142                                                                              {
143                                                                                \tl_set:Nn \l__scripture_c
144                                                                                  { vc }
145                                                                              }
146                                                                            { top }
147                                                                              {
148                                                                                \tl_set:Nn \l__scripture_c
149                                                                                  { t }
150                                                                              }
151                                                                          }
152                                                                        }
153        , chapter / valign          .value_required:n   = true
154        , chapter / valign          .initial:n          = bottom
155        , chapter / xchar           .tl_set:N           = \l__scripture_X_char_tl
156        , chapter / xchar           .value_required:n   = true
157        , chapter / xchar           .initial:n          = X
158        , defaults                  .code:n             = \cs_if_exist_use:N \__scripture_setu
159        , defaults                  .value_forbidden:n  = true
160        , extraskip                 .tl_set:N           = \l__scripture_extraskip_tl
161        , extraskip                 .value_required:n   = true
162        , extraskip                 .initial:n          = \medskipamount
163        , font                      .tl_set:N           = \l__scripture_font_tl
164        , font                      .value_required:n   = true
165        , font                      .initial:n          =
166        , heading / aboveskip       .tl_set:N           = \l__scripture_heading_aboveskip_tl
167        , heading / aboveskip       .value_required:n   = true
168        , heading / aboveskip       .initial:n          = \bigskipamount
169        , heading / afterindent     .bool_set:N         = \l__scripture_heading_afterindent_bo
170        , heading / afterindent     .default:n          = true
171        , heading / afterindent     .initial:n          = false
172        , heading / align           .choices:nn         = { left, right, center }
173                                                            {
174                                                              \tl_if_exist:NF \l__scripture_head
175                                                                { \tl_new:N \l__scripture_headin
176                                                              \str_case:Vn \l_keys_choice_tl
177                                                                {
178                                                                  { left }
179                                                                    {
180                                                                      \tl_set:Nn \l__scripture_h
181                                                                        { \raggedright }
182                                                                    }
183                                                                  { right }
184                                                                    {
185                                                                      \tl_set:Nn \l__scripture_h
186                                                                        { \raggedleft }
187                                                                    }
188                                                                  { center }
```

34

```
189                                                          {
190                                                            \tl_set:Nn \l__scripture_h
191                                                              { \centering }
192                                                          }
193                                                        }
194                                                      }
195      , heading / align              .value_required:n  = true
196      , heading / align              .initial:n         = left
197      , heading / belowskip          .tl_set:N          = \l__scripture_heading_belowskip_tl
198      , heading / belowskip          .value_required:n  = true
199      , heading / belowskip          .initial:n         = \medskipamount
200      , heading / font               .tl_set:N          = \l__scripture_heading_font_tl
201      , heading / font               .value_required:n  = true
202      , heading / font               .initial:n         = \small\itshape
203      , heading / format             .cs_set:Np         = \__scripture_heading_format:n #1
204      , heading / format             .value_required:n  = true
205      , heading / format             .initial:n         = #1
206      , heading / hide               .bool_set_inverse:N = \l__scripture_heading_show_bool
207      , heading / hide               .default:n         = true
208      , heading / show               .bool_set:N        = \l__scripture_heading_show_bool
209      , heading / show               .default:n         = true
210      , heading / show               .initial:n         = true
211      , indent                       .bool_set:N        = \l__scripture_indent_bool
212      , indent                       .default:n         = true
213      , indent                       .initial:n         = true
214      , inline                       .bool_set:N        = \l__scripture_inline_bool
215      , inline                       .default:n         = true
216      , inline                       .initial:n         = false
217      , inline / begin               .tl_set:N          = \l__scripture_inline_begin_tl
218      , inline / begin               .value_required:n  = true
219      , inline / begin               .initial:n         = ``
220      , inline / end                 .tl_set:N          = \l__scripture_inline_end_tl
221      , inline / end                 .value_required:n  = true
222      , inline / end                 .initial:n         = ''
223      , inline / reference / format  .cs_set:Np         = \__scripture_inline_ref_format:n #1
224      , inline / reference / format  .value_required:n  = true
225      , inline / reference / format  .initial:n         = (#1)
226      , inline / reference / sep     .tl_set:N          = \l__scripture_inline_ref_sep_tl
227      , inline / reference / sep     .value_required:n  = true
228      , inline / reference / sep     .initial:n         = 0.5em
229      , inline / version / delim     .tl_set:N          = \l__scripture_inline_version_delim_t
230      , inline / version / delim     .value_required:n  = true
231      , inline / version / delim     .initial:n         = \c_space_tl
232      , inline / version / format    .cs_set:Np         = \__scripture_inline_version_format:n
233      , inline / version / format    .value_required:n  = true
234      , inline / version / format    .initial:n         = #1
235      , language                     .tl_set:N          = \l__scripture_language_tl
236      , language                     .value_required:n  = true
237      , language                     .initial:n         =
238      , language / variant           .tl_set:N          = \l__scripture_language_variant_tl
239      , language / variant           .value_required:n  = true
240      , language / variant           .initial:n         =
241      , leftmargin                   .tl_set:N          = \l__scripture_leftmargin_tl
242      , leftmargin                   .value_required:n  = true
```

35

```
243      , leftmargin                .initial:n         = \c_zero_dim
244      , name / font               .tl_set:N          = \l__scripture_name_font_tl
245      , name / font               .value_required:n  = true
246      , name / font               .initial:n         = \scshape
247      , name / format             .cs_set:Np         = \__scripture_name_format:n #1
248      , name / format             .value_required:n  = true
249      , name / format             .initial:n         = #1
250      , noindent                  .bool_set_inverse:N = \l__scripture_indent_bool
251      , noindent                  .default:n         = true
252      , parindent                 .tl_set:N          = \l__scripture_parindent_tl
253      , parindent                 .value_required:n  = true
254      , parindent                 .initial:n         = \parindent
255      , parskip                   .tl_set:N          = \l__scripture_parskip_tl
256      , parskip                   .value_required:n  = true
257      , parskip                   .initial:n         = \parskip
258      , pilcrow                   .bool_set:N        = \l__scripture_pilcrow_bool
259      , pilcrow                   .default:n         = true
260      , pilcrow                   .initial:n         = false
261      , pilcrow / sep             .tl_set:N          = \l__scripture_pilcrow_sep_tl
262      , pilcrow / sep             .value_required:n  = true
263      , pilcrow / sep             .initial:n         = 0.25em
264      , redletter                 .bool_set:N        = \l__scripture_red_letter_bool
265      , redletter                 .default:n         = true
266      , redletter                 .initial:n         = false
267      , redletter / colour        .tl_set:N          = \l__scripture_red_letter_colour_tl
268      , redletter / colour        .value_required:n  = true
269      , redletter / colour        .initial:n         = red!80!black
270      , redletter / color         .meta:n            = { redletter / colour = { #1 } }
271      , reference / align         .choices:nn        = { left, right }
272                                                       {
273                                                         \tl_if_exist:NF \l__scripture_ref_
274                                                           { \tl_new:N \l__scripture_ref_al
275                                                         \tl_set_eq:NN \l__scripture_ref_al
276                                                       }
277      , reference / align         .value_required:n  = true
278      , reference / align         .initial:n         = right
279      , reference / colour        .tl_set:N          = \l__scripture_ref_colour_tl
280      , reference / colour        .value_required:n  = true
281      , reference / colour        .initial:n         = .
282      , reference / color         .meta:n            = { reference / colour = { #1 } }
283      , reference / font          .tl_set:N          = \l__scripture_ref_font_tl
284      , reference / font          .value_required:n  = true
285      , reference / font          .initial:n         = \bfseries
286      , reference / format        .cs_set:Np         = \__scripture_ref_format:n #1
287      , reference / format        .value_required:n  = true
288      , reference / format        .initial:n         = #1
289      , reference / newline       .meta:n            = { reference / sep = \linewidth }
290      , reference / newline       .value_forbidden:n = true
291      , reference / position      .choices:nn        = { start, end }
292                                                       {
293                                                         \tl_if_exist:NF \l__scripture_ref_
294                                                           { \tl_new:N \l__scripture_ref_po
295                                                         \tl_set_eq:NN \l__scripture_ref_po
296                                                       }
```

```
297       , reference / position       .value_required:n   = true
298       , reference / position       .initial:n          = end
299       , reference / sep            .tl_set:N           = \l__scripture_ref_sep_tl
300       , reference / sep            .value_required:n   = true
301       , reference / sep            .initial:n          = 2em
302       , reference / start / inline .bool_set_inverse:N = \l__scripture_ref_start_newline_bool
303       , reference / start / inline .default:n          = true
304       , reference / start / newline .bool_set:N        = \l__scripture_ref_start_newline_bool
305       , reference / start / newline .default:n         = true
306       , reference / start / newline .initial:n         = false
307       , reference / start / sep     .tl_set:N          = \l__scripture_ref_start_sep_tl
308       , reference / start / sep     .value_required:n  = true
309       , reference / start / sep     .initial:n         = 1em
310       , reference / start / vsep    .tl_set:N          = \l__scripture_ref_vertical_sep_tl
311       , reference / start / vsep    .value_required:n  = true
312       , reference / start / vsep    .initial:n         = 0pt
313       , rightmargin                 .tl_set:N          = \l__scripture_rightmargin_tl
314       , rightmargin                 .value_required:n  = true
315       , rightmargin                 .initial:n         = \c_zero_dim
316       , selah / text                .tl_set:N          = \l__scripture_selah_text_tl
317       , selah / text                .value_required:n  = true
318       , selah / text                .initial:n         = Selah
319       , selah / font                .tl_set:N          = \l__scripture_selah_font_tl
320       , selah / font                .value_required:n  = true
321       , selah / font                .initial:n         = \itshape
322       , selah / format              .cs_set:Np         = \__scripture_selah_format:n #1
323       , selah / format              .value_required:n  = true
324       , selah / format              .initial:n         = #1
325       , selah / sep                 .tl_set:N          = \l__scripture_selah_sep_tl
326       , selah / sep                 .value_required:n  = true
327       , selah / sep                 .initial:n         = 1em
328       , style                       .choice:
329       , style / unknown             .code:n            = \msg_error:nnx
330                                                            { scripture }
331                                                            { unknown-style }
332                                                            { \exp_not:n {#1} }
333       , textright / sep             .tl_set:N          = \l__scripture_text_right_sep_tl
334       , textright / sep             .value_required:n  = true
335       , textright / sep             .initial:n         = 1em
336       , verse / colour              .tl_set:N          = \l__scripture_verse_colour_tl
337       , verse / colour              .value_required:n  = true
338       , verse / colour              .initial:n         = .
339       , verse / color               .meta:n            = { verse / colour = { #1 } }
340       , verse / first               .bool_set:N        = \l__scripture_verse_first_bool
341       , verse / first               .default:n         = true
342       , verse / first               .initial:n         = false
343       , verse / firstformat         .cs_set:Np         = \__scripture_verse_first_format:n #1
344       , verse / firstformat         .value_required:n  = true
345       , verse / firstformat         .initial:n         = #1
346       , verse / firstsep            .tl_set:N          = \l__scripture_verse_first_sep_tl
347       , verse / firstsep            .value_required:n  = true
348       , verse / firstsep            .initial:n         = 0.5em
349       , verse / font                .tl_set:N          = \l__scripture_verse_font_tl
350       , verse / font                .value_required:n  = true
```

```
351      , verse / font              .initial:n         =
352      , verse / format            .cs_set:Np         = \__scripture_verse_format:n #1
353      , verse / format            .value_required:n  = true
354      , verse / format            .initial:n         = \textsuperscript{#1}
355      , verse / hide              .bool_set_inverse:N = \l__scripture_verse_show_bool
356      , verse / hide              .default:n         = true
357      , verse / para              .bool_set:N        = \l__scripture_verse_para_bool
358      , verse / para              .default:n         = true
359      , verse / para              .initial:n         = false
360      , verse / para / indent     .tl_set:N          = \l__scripture_para_indent_tl
361      , verse / para / indent     .value_required:n  = true
362      , verse / para / indent     .initial:n         = \l__scripture_parindent_tl
363      , verse / sep               .tl_set:N          = \l__scripture_verse_sep_tl
364      , verse / sep               .value_required:n  = true
365      , verse / sep               .initial:n         = 0.05em
366      , verse / show              .bool_set:N        = \l__scripture_verse_show_bool
367      , verse / show              .default:n         = true
368      , verse / show              .initial:n         = true
369      , version                   .tl_set:N          = \l__scripture_version_tl
370      , version                   .value_required:n  = true
371      , version                   .initial:n         =
372      , version / delim           .tl_set:N          = \l__scripture_version_delim_tl
373      , version / delim           .value_required:n  = true
374      , version / delim           .initial:n         = \c_space_tl
375      , version / format          .cs_set:Np         = \__scripture_version_format:n #1
376      , version / format          .value_required:n  = true
377      , version / format          .initial:n         = (#1)
378      , version / position        .choices:nn        = { withref, end }
379                                                       {
380                                                         \tl_if_exist:NF \l__scripture_vers
381                                                           { \tl_new:N \l__scripture_versio
382                                                         \tl_set_eq:NN \l__scripture_versio
383                                                       }
384      , version / position        .value_required:n  = true
385      , version / position        .initial:n         = withref
```

Options for the `center` environment.

```
386      , center / aboveskip        .tl_set:N          = \l__scripture_center_aboveskip_tl
387      , center / aboveskip        .value_required:n  = true
388      , center / aboveskip        .initial:n         = \medskipamount
389      , center / belowskip        .tl_set:N          = \l__scripture_center_belowskip_tl
390      , center / belowskip        .value_required:n  = true
391      , center / belowskip        .initial:n         = \medskipamount
392      , center / leftmargin       .tl_set:N          = \l__scripture_center_leftmargin_tl
393      , center / leftmargin       .value_required:n  = true
394      , center / leftmargin       .initial:n         = \c_zero_dim
395      , center / rightmargin      .tl_set:N          = \l__scripture_center_rightmargin_tl
396      , center / rightmargin      .value_required:n  = true
397      , center / rightmargin      .initial:n         = \c_zero_dim
```

Options for the `flushright` environment.

```
398      , flushright / aboveskip    .tl_set:N          = \l__scripture_flushright_aboveskip_t
399      , flushright / aboveskip    .value_required:n  = true
400      , flushright / aboveskip    .initial:n         = \medskipamount
401      , flushright / belowskip    .tl_set:N          = \l__scripture_flushright_belowskip_t
```

```
402        , flushright / belowskip      .value_required:n  = true
403        , flushright / belowskip      .initial:n         = \medskipamount
404        , flushright / leftmargin     .tl_set:N          = \l__scripture_flushright_leftmargin_
405        , flushright / leftmargin     .value_required:n  = true
406        , flushright / leftmargin     .initial:n         = \c_zero_dim
407        , flushright / rightmargin    .tl_set:N          = \l__scripture_flushright_rightmargin
408        , flushright / rightmargin    .value_required:n  = true
409        , flushright / rightmargin    .initial:n         = \c_zero_dim
```

Options for the hanging environment.

```
410        , hanging / aboveskip         .tl_set:N          = \l__scripture_hanging_aboveskip_tl
411        , hanging / aboveskip         .value_required:n  = true
412        , hanging / aboveskip         .initial:n         = \medskipamount
413        , hanging / belowskip         .tl_set:N          = \l__scripture_hanging_belowskip_tl
414        , hanging / belowskip         .value_required:n  = true
415        , hanging / belowskip         .initial:n         = \medskipamount
416        , hanging / chapter / indent  .bool_set:N        = \l__scripture_hanging_chapter_indent
417        , hanging / chapter / indent  .default:n         = true
418        , hanging / chapter / indent  .initial:n         = true
419        , hanging / hang              .tl_set:N          = \l__scripture_hanging_hang_tl
420        , hanging / hang              .value_required:n  = true
421        , hanging / hang              .initial:n         = 2em
422        , hanging / leftmargin        .tl_set:N          = \l__scripture_hanging_leftmargin_tl
423        , hanging / leftmargin        .value_required:n  = true
424        , hanging / leftmargin        .initial:n         = 1em
425        , hanging / rightmargin       .tl_set:N          = \l__scripture_hanging_rightmargin_tl
426        , hanging / rightmargin       .value_required:n  = true
427        , hanging / rightmargin       .initial:n         = \c_zero_dim
428        , hanging / verse / left      .bool_set_inverse:N = \l__scripture_hanging_verse_right_bo
429        , hanging / verse / left      .default:n         = true
430        , hanging / verse / right     .bool_set:N        = \l__scripture_hanging_verse_right_bo
431        , hanging / verse / right     .default:n         = true
432        , hanging / verse / right     .initial:n         = true
433        , hanging / verse / sep       .tl_set:N          = \l__scripture_hanging_verse_sep_tl
434        , hanging / verse / sep       .value_required:n  = true
435        , hanging / verse / sep       .initial:n         = 0.05em
```

Options for the mid-paragraph chapters (including the midparachap environment).

```
436        , midparachap / hideverse     .bool_set_inverse:N = \l__scripture_midparachap_show_verse
437        , midparachap / hideverse     .default:n         = true
438        , midparachap / showverse     .bool_set:N        = \l__scripture_midparachap_show_verse
439        , midparachap / showverse     .default:n         = true
440        , midparachap / showverse     .initial:n         = true
```

Options for the narrow environment.

```
441        , narrow / aboveskip          .tl_set:N          = \l__scripture_narrow_aboveskip_tl
442        , narrow / aboveskip          .value_required:n  = true
443        , narrow / aboveskip          .initial:n         = \medskipamount
444        , narrow / belowskip          .tl_set:N          = \l__scripture_narrow_belowskip_tl
445        , narrow / belowskip          .value_required:n  = true
446        , narrow / belowskip          .initial:n         = \medskipamount
447        , narrow / leftmargin         .tl_set:N          = \l__scripture_narrow_leftmargin_tl
448        , narrow / leftmargin         .value_required:n  = true
449        , narrow / leftmargin         .initial:n         = 1em
450        , narrow / rightmargin        .tl_set:N          = \l__scripture_narrow_rightmargin_tl
```

```
451       , narrow / rightmargin        .value_required:n   = true
452       , narrow / rightmargin        .initial:n          = 1em
```

Options for the `poetry` environment.

```
453       , poetry / aboveskip          .tl_set:N           = \l__scripture_poetry_aboveskip_tl
454       , poetry / aboveskip          .value_required:n   = true
455       , poetry / aboveskip          .initial:n          = \medskipamount
456       , poetry / belowskip          .tl_set:N           = \l__scripture_poetry_belowskip_tl
457       , poetry / belowskip          .value_required:n   = true
458       , poetry / belowskip          .initial:n          = \medskipamount
459       , poetry / bigindent          .tl_set:N           = \l__scripture_poetry_big_indent_tl
460       , poetry / bigindent          .value_required:n   = true
461       , poetry / bigindent          .initial:n          = 4em
462       , poetry / indent             .tl_set:N           = \l__scripture_poetry_indent_tl
463       , poetry / indent             .value_required:n   = true
464       , poetry / indent             .initial:n          = 1em
465       , poetry / leftmargin         .tl_set:N           = \l__scripture_poetry_leftmargin_tl
466       , poetry / leftmargin         .value_required:n   = true
467       , poetry / leftmargin         .initial:n          = 1em
468       , poetry / rightmargin        .tl_set:N           = \l__scripture_poetry_rightmargin_tl
469       , poetry / rightmargin        .value_required:n   = true
470       , poetry / rightmargin        .initial:n          = \c_zero_dim
471       , poetry / verse / left       .bool_set_inverse:N = \l__scripture_poetry_verse_right_boo
472       , poetry / verse / left       .default:n          = true
473       , poetry / verse / right      .bool_set:N         = \l__scripture_poetry_verse_right_boo
474       , poetry / verse / right      .default:n          = true
475       , poetry / verse / right      .initial:n          = true
476       , poetry / verse / sep        .tl_set:N           = \l__scripture_poetry_verse_sep_tl
477       , poetry / verse / sep        .value_required:n   = true
478       , poetry / verse / sep        .initial:n          = 0.05em
479     }
```

Process package options.

```
480   \ProcessKeyOptions
```

Create interface to reset all keys to defaults.

```
481   \tl_new:N \l__scripture_default_options_tl
482   \keys_precompile:nnN { scripture }
483     {
484       , aboveskip                 = \c_zero_skip
485       , added / font              =
486       , added / format            = \emph { #1 }
487       , after                     =
488       , before                    =
489       , belowskip                 = \c_zero_skip
490       , colour                    = .
491       , compact                   = false
492       , chapter / align           = left
493       , chapter / font            = \bfseries
494       , chapter / colour          = .
495       , chapter / drop            = true
496       , chapter / format          = #1
497       , chapter / para            = false
498       , chapter / para / aboveskip = \bigskipamount
499       , chapter / para / belowskip = \medskipamount
```

```
500    , chapter / para / indent   = true
501    , chapter / sep             = 0.5em
502    , chapter / show            = true
503    , chapter / showverse       = false
504    , chapter / smash           = false
505    , chapter / valign          = bottom
506    , chapter / xchar           = X
507    , extraskip                 = \medskipamount
508    , font                      =
509    , heading / aboveskip       = \bigskipamount
510    , heading / afterindent     = false
511    , heading / align           = left
512    , heading / belowskip       = \medskipamount
513    , heading / font            = \small\itshape
514    , heading / format          = #1
515    , heading / show            = true
516    , indent                    = true
517    , inline                    = false
518    , inline / begin            = ``
519    , inline / end              = ''
520    , inline / reference / format = (#1)
521    , inline / reference / sep  = 0.5em
522    , inline / version / delim  = \c_space_tl
523    , inline / version / format = #1
524    , language                  =
525    , language / variant        =
526    , leftmargin                = \c_zero_dim
527    , name / font               = \scshape
528    , name / format             = #1
529    , parindent                 = \parindent
530    , parskip                   = \parskip
531    , pilcrow                   = false
532    , pilcrow / sep             = 0.25em
533    , redletter                 = false
534    , redletter / colour        = red!80!black
535    , reference / align         = right
536    , reference / colour        = .
537    , reference / font          = \bfseries
538    , reference / format        = #1
539    , reference / position      = end
540    , reference / sep           = 2em
541    , reference / start / newline = false
542    , reference / start / sep   = 1em
543    , reference / start / vsep  = 0pt
544    , rightmargin               = \c_zero_dim
545    , selah / text              = Selah
546    , selah / font              = \itshape
547    , selah / format            = #1
548    , selah / sep               = 1em
549    , textright / sep           = 1em
550    , verse / colour            = .
551    , verse / first             = false
552    , verse / firstformat       = #1
553    , verse / firstsep          = 0.5em
```

```
554      , verse / font                =
555      , verse / format              = \textsuperscript{#1}
556      , verse / para                = false
557      , verse / para / indent       = \l__scripture_parindent_tl
558      , verse / sep                 = 0.05em
559      , verse / show                = true
560      , version                     =
561      , version / delim             = \c_space_tl
562      , version / format            = (#1)
563      , version / position          = withref
564      , center / aboveskip          = \medskipamount
565      , center / belowskip          = \medskipamount
566      , center / leftmargin         = \c_zero_dim
567      , center / rightmargin        = \c_zero_dim
568      , flushright / aboveskip      = \medskipamount
569      , flushright / belowskip      = \medskipamount
570      , flushright / leftmargin     = \c_zero_dim
571      , flushright / rightmargin    = \c_zero_dim
572      , hanging / aboveskip         = \medskipamount
573      , hanging / belowskip         = \medskipamount
574      , hanging / chapter / indent  = true
575      , hanging / hang              = 2em
576      , hanging / leftmargin        = 1em
577      , hanging / rightmargin       = \c_zero_dim
578      , hanging / verse / right     = true
579      , hanging / verse / sep       = 0.05em
580      , midparachap / showverse     = true
581      , narrow / aboveskip          = \medskipamount
582      , narrow / belowskip          = \medskipamount
583      , narrow / leftmargin         = 1em
584      , narrow / rightmargin        = 1em
585      , poetry / aboveskip          = \medskipamount
586      , poetry / belowskip          = \medskipamount
587      , poetry / bigindent          = 4em
588      , poetry / indent             = 1em
589      , poetry / leftmargin         = 1em
590      , poetry / rightmargin        = \c_zero_dim
591      , poetry / verse / right      = true
592      , poetry / verse / sep        = 0.05em
593    }
594    \l__scripture_default_options_tl
595  \cs_new_protected:Nn \__scripture_setup_reset_defaults:
596    {
597      \l__scripture_default_options_tl
598    }
```

__scripture_setup_style:nnn Specify options which can later be activated using the style option. If the first argument is true, then the options are appended rather than replaced.

```
599  \cs_new_protected:Nn \__scripture_setup_style:nnn
600    {
601      \bool_if:NTF #1
602        {
603          \tl_if_exist:cF { g__scripture_style_ #2 _tl }
604            { \tl_new:c { g__scripture_style_ #2 _tl } }
```

```
605         \__scripture_setup_style:nne \c_false_bool { #2 }
606           { \tl_use:c { g__scripture_style_ #2 _tl } , #3 }
607       }
608       {
609         \tl_gset:cn { g__scripture_style_ #2 _tl } { #3 }
610         \keys_define:nn { scripture }
611           { style / #2 .meta:n = { #3 } }
612       }
613   }
614 \cs_generate_variant:Nn \__scripture_setup_style:nnn { nne }
```

(*End of definition for* `\__scripture_setup_style:nnn`.)

3.4 Hooks

<div style="text-align: right">scripture/chap/before
scripture/chap/after
scripture/verse/before
scripture/verse/after</div>

Hooks for before and after chapters and verses. The current chapter and verse is available in the hooks as `\scripturecurrentchapter` and `\scripturecurrentverse` respectively. These are global.

```
615 \hook_new_pair:nn { scripture / chap / before } { scripture / chap / after }
616 \hook_new_pair:nn { scripture / verse / before } { scripture / verse / after }
```

(*End of definition for* `scripture/chap/before` *and others. These variables are documented on page 11.*)

<div style="text-align: right">scripture/pilcrow
scripture/poetry/pilcrow</div>

These are internal hooks used to insert a pilcrow at the beginning of a paragraph. Pilcrows are activated using the `pilcrow` option and are only in effect when the `verse/para` and/or `compact` options are in use.

```
617 \hook_new:n { scripture / pilcrow }
618 \hook_new:n { scripture / poetry / pilcrow }
```

(*End of definition for* `scripture/pilcrow` *and* `scripture/poetry/pilcrow`. *These variables are documented on page* **??**.)

3.5 The main `scripture` environment

<div style="text-align: right">\l__scripture_active_bool
\l__scripture_active_inner_bool</div>

Flags to ensure that environments are not nested.

```
619 \bool_new:N \l__scripture_active_bool
620 \bool_new:N \l__scripture_active_inner_bool
```

(*End of definition for* `\l__scripture_active_bool` *and* `\l__scripture_active_inner_bool`.)

<div style="text-align: right">\g__scripture_mode_vertical_inner_bool</div>

Whether an inner environment is starting in vertical mode. This flag is only used with the `compact` and `verse/para` options.

```
621 \bool_new:N \g__scripture_mode_vertical_inner_bool
```

(*End of definition for* `\g__scripture_mode_vertical_inner_bool`.)

<div style="text-align: right">\l__scripture_clubpenalty_saved_int</div>

Holds saved club penalty so we can change and restore it around chapter breaks.

```
622 \int_new:N \l__scripture_clubpenalty_saved_int
```

(*End of definition for* `\l__scripture_clubpenalty_saved_int`.)

<div style="text-align: right">\g__scripture_prev_inner_below_skip</div>

Save the last below skip of an inner environment so it can be removed if the main scripture environment ends with and inner environment.

```
623 \skip_new:N \g__scripture_prev_inner_below_skip
```

(End of definition for `\g__scripture_prev_inner_below_skip`.)

`\l_scripture_outer_itemindent_dim` Holds the `\itemindent` of the outer list. Used to calculate the `\itemindent` to retain correct indenting when a list environment is the first thing in an outer list.

```
624 \dim_new:N \l__scripture_outer_itemindent_dim
```

(End of definition for `\l__scripture_outer_itemindent_dim`.)

`\__scripture_setup_list_noindent:n` Set `\noindent` to do a negative horizontal skip a list is the first item in a list then restore it at the end of the paragraph. This allows `\itemindent` to be set to a negative value and still have `\noindent` behave as expected.

```
625 \cs_new_protected_nopar:Nn \__scripture_setup_list_noindent:n
626   {
627     \legacy_if:nT { @newlist }
628       {
629         \cs_set_protected_nopar:Npn \noindent
630           {
631             \skip_horizontal:n { -#1 }
632             \bool_if:NT \l__scripture_verse_para_bool
633               {
634                 \peek_meaning:NTF \vs
635                   {
636                     \bool_gset_true:N \g__scripture_suppress_next_verse_para_bool
637                     \__scripture_set_pilcrow_hook:
638                   }
639                   {
640                     \str_if_eq:VnF \l__scripture_currenvir_str { poetry }
641                       { \__scripture_pilcrow_output: }
642                   }
643               }
644           }
645         \hook_gput_next_code:nn { para / end }
646           {
647             \bool_if:NTF \l__scripture_verse_para_bool
648               {
649                 \str_if_eq:VnTF \l__scripture_currenvir_str { poetry }
650                   { \cs_set_eq:NN \noindent \tex_noindent:D }
651                   { \cs_set_eq:NN \noindent \__scripture_para_noindent: }
652               }
653               { \cs_set_eq:NN \noindent \tex_noindent:D }
654           }
655       }
656   }
```

(End of definition for `\__scripture_setup_list_noindent:n`.)

`\__scripture_skip_diff:NN` Find the difference between two skips while avoiding negative stretch and shrink components.

```
657 \cs_new_protected:Nn \__scripture_skip_diff:NN
658   {
659     \skip_eval:n
660       {
661         \dim_eval:n { \dimexpr #1 - \dimexpr #2 }
662         ~ plus ~
```

44

```
663         \dim_max:nn { \gluestretch #1 - \gluestretch #2 } \c_zero_dim
664       ~ minus ~
665         \dim_max:nn { \glueshrink #1 - \glueshrink #2 } \c_zero_dim
666     }
667   }
```

(*End of definition for* `\__scripture_skip_diff:NN`.)

`\__scripture_extra_skip:` Insert a little extra vertical space between two paragraphs. This function tests if the previous paragraph is only one line in length and began with a `\ch` function. If this condition is met, more space is added if necessary to avoid the following paragraph clashing with the drop chapter.

```
668 \cs_new_protected:Nn \__scripture_extra_skip:
669   {
670     \mode_if_vertical:F \para_end:
671     \int_compare:nNnT
672       \g__scripture_chap_par_prevgraf_int = 1
673       {
674         \dim_compare:nNnT
675           { \parskip + \glueexpr \l__scripture_extraskip_tl } < \baselineskip
676           {
677             \addvspace { \baselineskip }
678           }
679         \int_gzero:N \g__scripture_chap_par_prevgraf_int
680       }
681     \addvspace { \l__scripture_extraskip_tl }
682   }
```

(*End of definition for* `\__scripture_extra_skip:`.)

`\__scripture_text_right:nn` Align a word to the right and finish a paragraph, pushing to the next line if there is not space for the word.

```
683 \cs_new_protected:Nn \__scripture_text_right:nn
684   {
685     \bool_lazy_or:nnTF
686       { \str_if_eq_p:Vn \l__scripture_currenvir_str { flushright } }
687       { \str_if_eq_p:Vn \l__scripture_currenvir_str { center } }
688       {
689         \unskip
690         \break
691         \strut
692         \hfill
693         #2
694         \para_end:
695       }
696       {
697         {
698           \unskip
699           \hfil
700           \penalty 50
701           \skip_horizontal:n { #1 }
702           \hbox:n {}
703           \nobreak
704           \hfill
```

```
705            #2
706            \skip_zero:N \parfillskip
707            \int_zero:N \finalhyphendemerits
708            \para_end:
709          }
710        }
711    }
```

(*End of definition for* `\__scripture_text_right:nn`.)

3.5.1 Formatting and printing quotation references

`\l__scripture_ref_box` Box to hold the formatted reference.

```
712 \box_new:N \l__scripture_ref_box
```

(*End of definition for* `\l__scripture_ref_box`.)

`\__scripture_format_full_ref:nn` Format a scripture quotation with its reference.

```
713 \cs_new_protected:Nn \__scripture_format_full_ref:nn
714    {
715      \group_begin:
716      \l__scripture_ref_font_tl
717      \str_if_eq:VnF \l__scripture_ref_colour_tl { . }
718        { \exp_args:NV \color_select:n \l__scripture_ref_colour_tl }
719      \__scripture_ref_format:n
720        {
721          #1
```

Append a Bible version if it has been specified.

```
722          \tl_if_empty:oF { #2 }
723            {
724              \tl_if_empty:oF { #1 }
725                { \l__scripture_version_delim_tl }
726              \__scripture_version_format:n { #2 }
727            }
728        }
729      \group_end:
730    }
```

(*End of definition for* `\__scripture_format_full_ref:nn`.)

`\__scripture_reference:n` Output the reference for the Scripture quotation.

```
731 \cs_new_protected:Nn \__scripture_reference:n
732    {
733      \hbox_set:Nn \l__scripture_ref_box
734        {
735          \__scripture_format_full_ref:nn { #1 } \l__scripture_version_tl
736        }
737      \dim_compare:nNnT
738        { \box_wd:N \l__scripture_ref_box + \g__scripture_final_line_dim + \l__scripture_ref_s
739        >
740        \linewidth
741        {
742          \skip_vertical:N \baselineskip
743          \tl_if_eq:NnT \l__scripture_ref_align_tl { left }
```

```
744                 {
745                   \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
746                     { \skip_vertical:N \baselineskip }
747                 }
748             }
749         \tl_if_eq:NnTF \l__scripture_ref_align_tl { right }
750           { \hfill }
751           {
752             \dim_compare:nNnT
753               { \box_wd:N \l__scripture_ref_box + \g__scripture_final_line_dim + \l__scripture_r
754               <
755               \linewidth
756               { \skip_horizontal:n { \g__scripture_final_line_dim + \l__scripture_ref_sep_tl } }
757           }
758         \box_use:N \l__scripture_ref_box
759       }
```

(*End of definition for* __scripture_reference:n.)

__scripture_reference_start_inline: Output the reference for the Scripture quotation when inline at the start.

```
760 \cs_new_protected:Nn \__scripture_reference_start_inline:
761   {
762     \tl_if_empty:NF \l__scripture_ref_tl
763       {
764         \tl_if_eq:NnTF \l__scripture_version_position_tl { withref }
765           { \__scripture_format_full_ref:nn { \l__scripture_ref_tl } { \l__scripture_version
766           { \__scripture_format_full_ref:nn { \l__scripture_ref_tl } { } }
767         \skip_horizontal:N \l__scripture_ref_start_sep_tl
768       }
769   }
```

(*End of definition for* __scripture_reference_start_inline:.)

__scripture_reference_start_newline: Output the reference for the Scripture quotation when on a new line at the start.

```
770 \cs_new_protected:Nn \__scripture_reference_start_newline:
771   {
772     \tl_if_empty:NF \l__scripture_ref_tl
773       {
774         \bool_gset_eq:NN \g_tmpa_bool \g__scripture_suppress_next_pilcrow_bool
775         \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
776         \noindent
777         \bool_gset_eq:NN \g__scripture_suppress_next_pilcrow_bool \g_tmpa_bool
778         \str_if_eq:VnF \l__scripture_currenvir_str { scripture }
779           { \skip_horizontal:n { -\leftmargin } }
780         \tl_if_eq:NnTF \l__scripture_version_position_tl { withref }
781           { \__scripture_format_full_ref:nn { \l__scripture_ref_tl } { \l__scripture_version
782           { \__scripture_format_full_ref:nn { \l__scripture_ref_tl } { } }
783         \bool_lazy_or:nnT
784           { \str_if_eq_p:Vn \l__scripture_currenvir_str { flushright } }
785           { \str_if_eq_p:Vn \l__scripture_currenvir_str { center } }
786           {
787             \hfill
788             \strut
789           }
790         \bool_if:NTF \l__scripture_compact_bool
```

```
791            { \para_end: }
792            {
793              \bool_lazy_all:nTF
794                {
795                  { \l__scripture_verse_para_bool }
796                  { \bool_not_p:n { \str_if_eq_p:Vn \l__scripture_currenvir_str { hanging } }
797                  { \bool_not_p:n { \str_if_eq_p:Vn \l__scripture_currenvir_str { poetry } } }
798                }
799                { \para_end: }
800                { \par }
801            }
802          \nobreak
803          \@afterheading
804          \addvspace { \l__scripture_ref_vertical_sep_tl }
805        }
806    }
```

(*End of definition for* `\__scripture_reference_start_newline:`*.*)

`\__scripture_reference_start_peek:`   Peek ahead at the beginning of the `scripture` environment to control how references are inserted at the start of a quotation.

```
807 \cs_new_protected:Nn \__scripture_reference_start_peek:
808    {
809      \peek_remove_spaces:n
810        {
811          \bool_lazy_and:nnTF
812            { \str_if_eq_p:Vn \l__scripture_ref_position_tl { start } }
813            { \bool_not_p:n { \tl_if_empty_p:N \l__scripture_ref_tl } }
814            {
815              \bool_lazy_or:nnF
816                { \str_if_eq_p:Vn \l__scripture_currenvir_str { scripture } }
817                { \str_if_eq_p:Vn \l__scripture_currenvir_str { midparachap } }
818                {
819                  \bool_if:NF \l__scripture_compact_bool
820                    { \bool_set_true:N \l__scripture_ref_start_newline_bool }
821                }
822              \peek_analysis_map_inline:n
823                {
824                  \bool_case:nF
825                    {
826                      { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \heading }
827                        { \__scripture_reference_start_newline: }
828                      { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \noindent }
829                        {
830                          \bool_if:NTF \l__scripture_ref_start_newline_bool
831                            { \__scripture_reference_start_newline: }
832                            { \__scripture_reference_start_inline: }
833                        }
834                      { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \nohang }
835                        {
836                          \bool_if:NTF \l__scripture_ref_start_newline_bool
837                            { \__scripture_reference_start_newline: }
838                            { \__scripture_reference_start_inline: }
839                        }
```

```
                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \nopilcrow }
                            {
                              \bool_if:NTF \l__scripture_ref_start_newline_bool
                                { \__scripture_reference_start_newline: }
                                { \__scripture_reference_start_inline: }
                              \peek_analysis_map_break:n
                                {
                                  ##1
                                  \__scripture_fix_protrusion:n { }
                                }
                            }
                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \begin }
                            {
                              \bool_if:NF \l__scripture_indent_bool
                                {
                                  \hook_gput_next_code:nn { para / begin }
                                    {
                                      \bool_lazy_or:nnT
                                        { \str_if_eq_p:Vn \l__scripture_currenvir_str { midparacha
                                        { \str_if_eq_p:Vn \l__scripture_currenvir_str { scripture
                                        { \para_omit_indent: }
                                    }
                                }
                            }
                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \ch }
                            {
                              \bool_if:NTF \l__scripture_ref_start_newline_bool
                                { \__scripture_reference_start_newline: }
                                {
                                  \cs_set_eq:NN \__scripture_temp_ch \ch
                                  \RenewDocumentCommand \ch { s o m }
                                    {
                                      \tl_if_novalue:nF { ##2 }
                                        {
                                          \tl_set:Nn \l__scripture_X_char_tl { ##2 }
                                        }
                                      \cs_set_eq:NN \ch \__scripture_temp_ch
                                    }
                                  \__scripture_reference_start_inline:
                                }
                            }
                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \vs }
                            {
                              \bool_if:NTF \l__scripture_ref_start_newline_bool
                                { \__scripture_reference_start_newline: }
                                {
                                  \cs_set_eq:NN \__scripture_temp_vs \vs
                                  \RenewDocumentCommand \vs { m }
                                    { \cs_set_eq:NN \vs \__scripture_temp_vs }
                                  \__scripture_reference_start_inline:
                                }
                            }
                        }
                        {
```

```
894                          \bool_if:NTF \l__scripture_ref_start_newline_bool
895                            {
896                              \__scripture_reference_start_newline:
897                              \bool_if:NTF \l__scripture_compact_bool
898                                {
899                                  \bool_if:NT \g__scripture_mode_vertical_inner_bool
900                                    { \__scripture_pilcrow_output: }
901                                }
902                                {
903                                  \str_if_eq:VnF \l__scripture_currenvir_str { poetry }
904                                  { \__scripture_pilcrow_output: }
905                                }
906                            }
907                            { \__scripture_reference_start_inline: }
908                        }
909                      \exp_args:No \token_if_eq_meaning:NNF { ##1 } \begin
910                        {
911                          \tl_clear:N \l__scripture_ref_tl
912                          \tl_if_eq:NnF \l__scripture_version_position_tl { end }
913                            { \tl_clear:N \l__scripture_version_tl }
914                        }
915                      \peek_analysis_map_break:n
916                        { \__scripture_fix_protrusion:o { ##1 } }
917                    }
918                }
919                {
920                  \peek_analysis_map_inline:n
921                    {
922                      \bool_case:nF
923                        {
924                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \heading }
925                          { }
926                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \noindent }
927                          { }
928                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \nohang }
929                          { }
930                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \nopilcrow }
931                            {
932                              \peek_analysis_map_break:n
933                                {
934                                  ##1
935                                  \__scripture_fix_protrusion:n { }
936                                }
937                            }
938                          { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \begin }
939                            {
940                              \bool_if:NF \l__scripture_indent_bool
941                                {
942                                  \hook_gput_next_code:nn { para / begin }
943                                    {
944                                      \bool_lazy_or:nnT
945                                        { \str_if_eq_p:Vn \l__scripture_currenvir_str { midparacha
946                                        { \str_if_eq_p:Vn \l__scripture_currenvir_str { scripture
947                                        { \para_omit_indent: }
```

```
948                              }
949                            }
950                          }
951                        { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \ch }
952                          { }
953                        { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \vs }
954                          {
955                            \mode_if_vertical:F
956                              { \__scripture_set_pilcrow_hook: }
957                          }
958                    }
959                    {
960                      \bool_if:NTF \l__scripture_compact_bool
961                        {
962                          \bool_if:NT \g__scripture_mode_vertical_inner_bool
963                            { \__scripture_pilcrow_output: }
964                        }
965                        {
966                          \str_if_eq:VnF \l__scripture_currenvir_str { poetry }
967                            { \__scripture_pilcrow_output: }
968                        }
969                    }
970                  \peek_analysis_map_break:n
971                    {
972                      \__scripture_fix_protrusion:o { ##1 }
973                    }
974                }
975            }
976        }
977    }
```

(*End of definition for* __scripture_reference_start_peek:.)

__scripture_compact_par: \par macro used when the **compact** option is in use.

```
978  \cs_new_protected:Nn \__scripture_compact_par:
979    {
980      \bool_gset_false:N \g__scripture_mode_vertical_inner_bool
981      \dim_set:Nn \parindent { \l__scripture_parindent_tl }
982      \peek_remove_spaces:n
983        {
984          \peek_analysis_map_inline:n
985            {
986              \bool_case:n
987                {
988                  { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \begin }
989                    { \bool_gset_true:N \g__scripture_mode_vertical_inner_bool }
990                  { \str_if_eq_p:Vn \l__scripture_currenvir_str { poetry } }
991                    { }
992                  { \exp_args:No \token_if_letter_p:N { ##1 } }
993                    { \__scripture_pilcrow_output: }
994                  { \exp_args:No \token_if_active_p:N { ##1 } }
995                    { \__scripture_pilcrow_output: }
996                  { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \vs }
997                    { \__scripture_set_pilcrow_hook: }
```

```
998                    }
999                \peek_analysis_map_break:n { ##1 }
1000            }
1001        }
1002    }
```

(*End of definition for* __scripture_compact_par:.)

__scripture_para_noindent: \noindent macro used when the verse/para option is in use.

```
1003 \cs_new_protected:Nn \__scripture_para_noindent:
1004    {
1005        \legacy_if:nTF { @newlist }
1006            {
1007                \tex_noindent:D
1008                \peek_meaning:NTF \vs
1009                    {
1010                        \bool_gset_true:N \g__scripture_suppress_next_verse_para_bool
1011                        \__scripture_set_pilcrow_hook:
1012                    }
1013                    { \__scripture_pilcrow_output: }
1014            }
1015            {
1016                \peek_meaning:NTF \vs
1017                    {
1018                        \bool_if:NT \l__scripture_verse_para_bool
1019                            { \para_end: }
1020                        \tex_noindent:D
1021                        \bool_gset_true:N \g__scripture_suppress_next_verse_para_bool
1022                        \__scripture_set_pilcrow_hook:
1023                    }
1024                    { \__scripture_pilcrow_output: }
1025            }
1026    }
```

(*End of definition for* __scripture_para_noindent:.)

\g__scripture_para_mode_vertical_bool Flag that __scripture_para_par: has been encountered and pilcrow not yet printed.

```
1027 \bool_new:N \g__scripture_para_mode_vertical_bool
```

(*End of definition for* \g__scripture_para_mode_vertical_bool.)

\g__scripture_post_env_indent_box Box to store the indent after an inner environment in the case where it is dropped by __scripture_dopeend:. This box is then inserted if the next thing is a \vs. Only used when the verse/para option is active.

```
1028 \box_new:N \g__scripture_post_env_indent_box
```

(*End of definition for* \g__scripture_post_env_indent_box.)

__scripture_para_par: \par macro used when the verse/para option is in use.

```
1029 \cs_new_protected:Nn \__scripture_para_par:
1030    {
1031        \bool_gset_true:N \g__scripture_para_mode_vertical_bool
1032        \dim_set:Nn \parindent { \l__scripture_parindent_tl }
1033        \peek_remove_spaces:n
1034            {
```

```
1035          \peek_analysis_map_inline:n
1036            {
1037              \bool_case:n
1038                {
1039                  { \exp_args:No \token_if_letter_p:N { ##1 } }
1040                    { \__scripture_pilcrow_output: }
1041                  { \exp_args:No \token_if_active_p:N { ##1 } }
1042                    { \__scripture_pilcrow_output: }
1043                  { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \vs }
1044                    { \__scripture_set_pilcrow_hook: }
1045                }
1046              \peek_analysis_map_break:n { ##1 }
1047            }
1048        }
1049    }
```

(*End of definition for* __scripture_para_par:*.*)

__scripture_doendpe: Custom version of \@doendpe macro used when the verse/para option is in use. This handles pilcrows after inner environments.

```
1050  \cs_new_protected:Nn \__scripture_doendpe:
1051    {
1052      \box_gclear:N \g__scripture_post_env_indent_box
1053      \legacy_if_set_true:n { @endpe }
1054      \cs_set_nopar:Npn \par
1055        {
1056          \@restorepar
1057          \clubpenalty \@clubpenalty
1058          \everypar { }
1059          \par
1060          \legacy_if_set_false:n { @endpe }
1061          \bool_gset_false:N \g__scripture_suppress_next_pilcrow_bool
1062          \__scripture_para_par:
1063        }
1064      \everypar
1065        {
1066          \box_gset_to_last:N \g__scripture_post_env_indent_box
1067          \hook_gput_next_code:nn { para / after }
1068            { \box_gclear:N \g__scripture_post_env_indent_box }
1069          \everypar { }
1070          \legacy_if_set_false:n { @endpe }
1071          \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
1072          \bool_if:NT \l__scripture_verse_para_bool
1073            {
1074              \box_set_wd:Nn
1075                \g__scripture_post_env_indent_box
1076                { \l__scripture_para_indent_tl }
1077            }
1078        }
1079    }
```

(*End of definition for* __scripture_doendpe:*.*)

__scripture_fix_protrusion:n   The use of \peek macros breaks protrusion at the beginning of list environments. This
__scripture_fix_protrusion:o   function checks conditions and add \leftprotrusion.

```
1080 \cs_new_protected:Nn \__scripture_fix_protrusion:n
1081   {
1082     \bool_if:NTF \l__scripture_compact_bool
1083       {
1084         \str_if_eq:VnT \l__scripture_currenvir_str { scripture }
1085           { \cs_if_exist_use:N \leftprotrusion }
1086       }
1087       { \cs_if_exist_use:N \leftprotrusion }
1088     #1
1089   }
1090 \cs_generate_variant:Nn \__scripture_fix_protrusion:n { o }
```

(*End of definition for* `\__scripture_fix_protrusion:n`*.*)

3.5.2 Red lettering

`\g__scripture_red_letter_active_bool` Flag indicating if currently in a red letter block of text.

```
1091 \bool_new:N \g__scripture_red_letter_active_bool
```

(*End of definition for* `\g__scripture_red_letter_active_bool`*.*)

`\__scripture_red_letter_on:` Mark the beginning of a section of text attributed to Jesus Christ.

```
1092 \cs_new_protected:Nn \__scripture_red_letter_on:
1093   {
1094     \bool_if:NT \l__scripture_red_letter_bool
1095       {
1096         \bool_gset_true:N \g__scripture_red_letter_active_bool
1097         \exp_args:NV \color_select:n \l__scripture_red_letter_colour_tl
1098         \str_if_eq:VnF \l__scripture_currenvir_str { scripture }
1099           {
1100             \hook_gput_next_code:nn { env / \l__scripture_currenvir_str / after }
1101               {
1102                 \exp_args:NV \color_select:n \l__scripture_red_letter_colour_tl
1103               }
1104           }
1105       }
1106   }
```

(*End of definition for* `\__scripture_red_letter_on:`*.*)

`\__scripture_red_letter_off:` Mark the beginning of a section of text attributed to Jesus Christ.

```
1107 \cs_new_protected:Nn \__scripture_red_letter_off:
1108   {
1109     \bool_if:NT \l__scripture_red_letter_bool
1110       {
1111         \color_select:n { scripture default colour }
1112         \bool_gset_false:N \g__scripture_red_letter_active_bool
1113         \str_if_eq:VnF \l__scripture_currenvir_str { scripture }
1114           {
1115             \hook_gput_next_code:nn { env / \l__scripture_currenvir_str / after }
1116               {
1117                 \color_select:n { scripture default colour }
1118               }
1119           }
1120       }
1121   }
```

(End of definition for `\__scripture_red_letter_off:`*.)*

3.5.3 Selecting language

`\__scripture_select_language:`   Select specified language if required.

```
1122 \cs_new_protected:Nn \__scripture_select_language:
1123   {
1124     \tl_if_empty:NF \l__scripture_language_tl
1125       {
1126         \@ifpackageloaded { babel }
1127           {
1128             \__scripture_set_textdirection:
1129             \selectlanguage { \l__scripture_language_tl }
1130           }
1131           {
1132             \@ifpackageloaded { polyglossia }
1133               {
1134                 \iflanguageloaded { \l__scripture_language_tl }
1135                   {
1136                     \tl_if_empty:NTF \l__scripture_language_variant_tl
1137                       {
1138                         \selectlanguage { \l__scripture_language_tl }
1139                       }
1140                       {
1141                         \selectlanguage
1142                           [ variant = \l__scripture_language_variant_tl ]
1143                           { \l__scripture_language_tl }
1144                       }
1145                   }
1146                   {
1147                     \msg_error:nnx
1148                       { scripture }
1149                       { language-error }
1150                       { \l__scripture_language_tl }
1151                   }
1152               }
1153               {
1154                 \msg_warning:nnx
1155                   { scripture }
1156                   { language-warning }
1157                   { \l__scripture_language_tl }
1158               }
1159           }
1160       }
1161   }
```

(End of definition for `\__scripture_select_language:`*.)*

3.5.4 Formatting and printing chapter numbers

`\l__scripture_chap_box`   Boxes use to store and calculate drop chapter.
`l__scripture_chap_tmp_box`
```
1162 \box_new:N \l__scripture_chap_box
1163 \box_new:N \l__scripture_chap_tmp_box
```

55

(End of definition for `\l__scripture_chap_box` *and* `l__scripture_chap_tmp_box`.)

`\l_scripture_chap_fontsize_dim` Calculated font size of a drop chapter number.

```
1164 \dim_new:N \l__scripture_chap_fontsize_dim
```

(End of definition for `\l__scripture_chap_fontsize_dim`.)

`\l_scripture_chap_height_dim` Calculated height of a drop chapter number.

```
1165 \dim_new:N \l__scripture_chap_height_dim
```

(End of definition for `\l__scripture_chap_height_dim`.)

`\g__scripture_chap_width_dim` Calculated width of drop chapter number.

```
1166 \dim_new:N \g__scripture_chap_width_dim
```

(End of definition for `\g__scripture_chap_width_dim`.)

`\g_scripture_chap_par_prevgraf_int` Store the number of lines a chapter paragraph takes up.

```
1167 \int_new:N \g__scripture_chap_par_prevgraf_int
```

(End of definition for `\g__scripture_chap_par_prevgraf_int`.)

`\l_scripture_chap_X_height_dim` Calculated height of X at current font size. This is used as the height for a drop chapter.

```
1168 \dim_new:N \l__scripture_chap_X_height_dim
```

(End of definition for `\l__scripture_chap_X_height_dim`.)

`\l_scripture_para_chap_indent_dim` Indent used to center chapters when placed in their own paragraph.

```
1169 \dim_new:N \l__scripture_para_chap_indent_dim
```

(End of definition for `\l__scripture_para_chap_indent_dim`.)

`\l_scripture_nodrop_chap_coffin` Coffin to place non-drop chapters inside.

```
1170 \coffin_new:N \l__scripture_nodrop_chap_coffin
```

(End of definition for `\l__scripture_nodrop_chap_coffin`.)

`\l_scripture_nodrop_chap_voffset_dim` Vertical offset to use when setting non-drop chapters.

```
1171 \dim_new:N \l__scripture_nodrop_chap_voffset_dim
```

(End of definition for `\l__scripture_nodrop_chap_voffset_dim`.)

`\__scripture_get_X_height:` Calculate the height of `\l__scripture_X_char_tl` for vertical alignment of non-drop chapters.

```
1172 \cs_new_protected:Nn \__scripture_get_X_height:
1173   {
1174     \hbox_set:Nn \l__scripture_chap_tmp_box { \l__scripture_X_char_tl }
1175     \dim_set:Nn \l__scripture_chap_X_height_dim
1176       { \box_ht:N \l__scripture_chap_tmp_box }
1177   }
```

(End of definition for `\__scripture_get_X_height:`.)

56

`\__scripture_drop_chap_set_up:n` Set up chapter drop cap and calculate its various dimensions.

```
1178 \cs_new_protected:Nn \__scripture_drop_chap_set_up:n
1179   {
1180     \cs_gset_nopar:Npn \scripturecurrentchapter { #1 }
1181     \cs_gset_nopar:Npn \scripturecurrentverse { 1 }
1182     \__scripture_get_X_height:
1183     \dim_set:Nn \l__scripture_chap_height_dim
1184       { \baselineskip + \l__scripture_chap_X_height_dim }
1185     \dim_set:Nn \l__scripture_chap_fontsize_dim
1186       {
1187         \f@size pt *
1188         \dim_ratio:nn
1189           { \l__scripture_chap_height_dim }
1190           { \l__scripture_chap_X_height_dim }
1191       }
1192     \hbox_set:Nn \l__scripture_chap_tmp_box
1193       {
1194         \l__scripture_chap_font_tl
1195         \fontsize
1196           { \l__scripture_chap_fontsize_dim }
1197           { \l__scripture_chap_fontsize_dim }
1198         \selectfont
1199         \str_if_eq:VnF \l__scripture_chapter_colour_tl { . }
1200           { \exp_args:NV \color_select:n \l__scripture_chapter_colour_tl }
1201         \__scripture_chap_format:n { #1 }
1202       }
1203     \dim_gset:Nn \g__scripture_chap_width_dim
1204       { \box_wd:N \l__scripture_chap_tmp_box }
```

Avoid breaks in the middle of drop chapters.

```
1205     \int_set_eq:NN \l__scripture_clubpenalty_saved_int \clubpenalty
1206     \clubpenalty 10000
1207     \bool_if:NF \l__scripture_verse_para_bool
1208       {
1209         \hook_gput_next_code:nn { para / after }
1210           {
1211             \int_gset_eq:NN \g__scripture_chap_par_prevgraf_int \prevgraf
1212             \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
1213               { \penalty 10000 }
1214             \hook_gput_next_code:nn { para / before }
1215               { \int_set_eq:NN \clubpenalty \l__scripture_clubpenalty_saved_int }
1216           }
1217       }
1218   }
```

(*End of definition for* `\__scripture_drop_chap_set_up:n`.)

`\__scripture_drop_chap_output:` Output a drop chapter.

```
1219 \cs_new_protected:Nn \__scripture_drop_chap_output:
1220   {
1221     \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
1222     \cs_if_eq:NNTF \noindent \__scripture_para_noindent:
1223       {
1224         \legacy_if:nTF { @newlist }
```

```
1225          {
1226            \skip_horizontal:n { -\l__scripture_parindent_tl }
1227          }
1228          { \tex_noindent:D }
1229        \__scripture_pilcrow_output:
1230      }
1231      { \noindent }
1232    \parshape 3 ~
1233      \dim_eval:n
1234        { \@totalleftmargin + \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl } ~
1235      \dim_eval:n
1236        { \linewidth - \g__scripture_chap_width_dim - \l__scripture_chap_sep_tl } ~
1237      \dim_eval:n
1238        { \@totalleftmargin + \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl } ~
1239      \dim_eval:n
1240        { \linewidth - \g__scripture_chap_width_dim - \l__scripture_chap_sep_tl } ~
1241      \@totalleftmargin ~
1242      \linewidth
1243    \group_begin:
1244    \hook_use:n { scripture / chap / before }
1245    \box_use:N \l__scripture_chap_box
1246    \hook_use:n { scripture / chap / after }
1247    \group_end:
1248    \bool_if:NT \l__scripture_chap_show_verse_bool
1249      { \__scripture_verse_output:n { 1 } }
1250    \bool_if:NT \l__scripture_verse_para_bool
1251      {
1252        \hook_gput_next_code:nn { para / after }
1253          {
1254            \int_gset_eq:NN \g__scripture_chap_par_prevgraf_int \prevgraf
1255            \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
1256              { \penalty 10000 }
1257            \hook_gput_next_code:nn { para / before }
1258              { \int_set_eq:NN \clubpenalty \l__scripture_clubpenalty_saved_int }
1259          }
1260      }
1261    \hook_gclear_next_code:n { scripture / pilcrow }
1262  }
```

(*End of definition for* `\__scripture_drop_chap_output:`.)

`\__scripture_nodrop_chap:n`   Format a non-drop chapter.

```
1263 \cs_new_protected:Nn \__scripture_nodrop_chap:n
1264   {
1265     \cs_gset_nopar:Npn \scripturecurrentchapter { #1 }
1266     \cs_gset_nopar:Npn \scripturecurrentverse { 1 }
1267     \group_begin:
1268     \bool_if:NT \l__scripture_verse_para_bool
1269       {
1270         \legacy_if:nF { @newlist }
1271           { \mode_if_vertical:F \para_end: }
1272       }
1273     \bool_if:NTF \l__scripture_chap_show_bool
1274       {
```

```
1275          \__scripture_get_X_height:
1276          \hcoffin_set:Nn \l__scripture_nodrop_chap_coffin
1277            {
1278              \str_if_eq:VnF \l__scripture_chapter_colour_tl { . }
1279                { \exp_args:NV \color_select:n \l__scripture_chapter_colour_tl }
1280              \l__scripture_chap_font_tl
1281              \__scripture_chap_format:n { #1 }
1282              \kern \l__scripture_chap_sep_tl
1283            }
1284          \dim_zero:N \l__scripture_nodrop_chap_voffset_dim
1285          \tl_if_eq:NnT \l__scripture_chapter_valign_tl { t }
1286            { \dim_set_eq:NN \l__scripture_nodrop_chap_voffset_dim \l__scripture_chap_X_height
1287          \tl_if_eq:NnT \l__scripture_chapter_valign_tl { vc }
1288            { \dim_set:Nn \l__scripture_nodrop_chap_voffset_dim { 0.5 \l__scripture_chap_X_hei
1289          \hook_use:n { scripture / chap / before }
1290          \bool_if:NTF \l__scripture_chap_smash_bool
1291            {
1292              \mode_leave_vertical:
1293              \smash
1294                {
1295                  \exp_args:NNoo
1296                    \coffin_typeset:Nnnnn \l__scripture_nodrop_chap_coffin
1297                      { \l__scripture_chapter_align_tl }
1298                      { \l__scripture_chapter_valign_tl }
1299                      { 0pt }
1300                      { \l__scripture_nodrop_chap_voffset_dim }
1301                }
1302            }
1303            {
1304              \exp_args:NNoo
1305                \coffin_typeset:Nnnnn \l__scripture_nodrop_chap_coffin
1306                  { \l__scripture_chapter_align_tl }
1307                  { \l__scripture_chapter_valign_tl }
1308                  { 0pt }
1309                  { \l__scripture_nodrop_chap_voffset_dim }
1310            }
1311          \nobreak
1312          \skip_horizontal:N \c_zero_skip
1313          \hook_use:n { scripture / chap / after }
1314          \bool_if:NT \l__scripture_chap_show_verse_bool
1315            {
1316              \__scripture_verse_output:n { 1 }
1317            }
1318          \str_if_eq:VnF \l__scripture_currenvir_str { midparachap }
1319            { \__scripture_pilcrow_output: }
1320        }
1321        {
1322          \hook_use:n { scripture / chap / before }
1323          \hook_use:n { scripture / chap / after }
1324          \vs { 1 }
1325        }
1326      \group_end:
1327      \parshape 1 ~ \@totalleftmargin ~ \linewidth
1328      \hook_gclear_next_code:n { scripture / pilcrow }
```

```
1329     }
```

(End of definition for __scripture_nodrop_chap:n.*)*

__scripture_para_chap:n   Format a chapter in its own paragraph.

```
1330 \cs_new_protected:Nn \__scripture_para_chap:n
1331   {
1332     \cs_gset_nopar:Npn \scripturecurrentchapter { #1 }
1333     \cs_gset_nopar:Npn \scripturecurrentverse { 1 }
1334     \group_begin:
1335     \skip_zero:N \parskip
1336     \bool_if:NTF \l__scripture_chap_show_bool
1337       {
1338         \dim_set_eq:NN \l__scripture_para_chap_indent_dim \linewidth
1339         \str_if_eq:VnTF \l__scripture_currenvir_str { scripture }
1340           {
1341             \dim_set:Nn \l__scripture_para_chap_indent_dim { 0.5 \l__scripture_para_chap_ind
1342           }
1343           {
1344             \dim_add:Nn \l__scripture_para_chap_indent_dim { \leftmargin + \rightmargin }
1345             \dim_set:Nn \l__scripture_para_chap_indent_dim { 0.5 \l__scripture_para_chap_ind
1346             \dim_sub:Nn \l__scripture_para_chap_indent_dim \leftmargin
1347           }
1348         \mode_if_vertical:F \para_end:
1349         \addvspace { \l__scripture_chap_para_aboveskip_tl }
1350         \bool_gset_eq:NN \g_tmpa_bool \g__scripture_suppress_next_pilcrow_bool
1351         \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
1352         \str_if_eq:VnTF \l__scripture_currenvir_str { midparachap }
1353           { \tex_noindent:D }
1354           { \noindent }
1355         \bool_gset_eq:NN \g__scripture_suppress_next_pilcrow_bool \g_tmpa_bool
1356         \skip_horizontal:N \l__scripture_para_chap_indent_dim
1357         \hbox_overlap_center:n
1358           {
1359             \hook_use:n { scripture / chap / before }
1360             \group_begin:
1361             \str_if_eq:VnF \l__scripture_chapter_colour_tl { . }
1362               { \exp_args:NV \color_select:n \l__scripture_chapter_colour_tl }
1363             \l__scripture_chap_font_tl
1364             \__scripture_chap_format:n { #1 }
1365             \group_end:
1366             \hook_use:n { scripture / chap / after }
1367           }
1368         \para_end:
1369         \nobreak
1370         \bool_lazy_any:nTF
1371           {
1372             { \str_if_eq_p:Vn \l__scripture_currenvir_str { scripture } }
1373             { \str_if_eq_p:Vn \l__scripture_currenvir_str { midparachap } }
1374             { \str_if_eq_p:Vn \l__scripture_currenvir_str { narrow } }
1375           }
1376           { \legacy_if_set:nn { @afterindent } { \l__scripture_chap_para_afterindent_bool }
1377           { \legacy_if_set_true:n { @afterindent } }
1378         \@afterheading
```

60

```
1379        }
1380        {
1381          \hook_use:n { scripture / chap / before }
1382          \hook_use:n { scripture / chap / after }
1383        }
1384     \peek_meaning:NTF \c_space_token
1385        {
1386          \addvspace { \l__scripture_chap_para_belowskip_tl }
1387          \group_end:
1388          \hook_use:n { scripture / poetry / pilcrow }
1389        }
1390        {
1391          \addvspace { \l__scripture_chap_para_belowskip_tl }
1392          \cs_if_eq:NNTF \vs \__scripture_poetry_mode_vertical_verse:n
1393             {
1394               \bool_gset_eq:NN \g_tmpa_bool \g__scripture_suppress_next_pilcrow_bool
1395               \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
1396               \noindent
1397               \vs { 1 }
1398               \bool_gset_eq:NN \g__scripture_suppress_next_pilcrow_bool \g_tmpa_bool
1399             }
1400             {
1401               \str_if_eq:VnT \l__scripture_currenvir_str { midparachap }
1402                 { \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool }
1403               \vs { 1 }
1404             }
1405          \group_end:
1406          \hook_use:n { scripture / poetry / pilcrow }
1407        }
1408    }
```

(*End of definition for* `\__scripture_para_chap:n`.)

`\__scripture_chap:n` Position a drop chapter.

```
1409 \cs_new_protected:Nn \__scripture_chap:n
1410   {
1411     \__scripture_drop_chap_set_up:n { #1 }
1412     \hbox_set:Nn \l__scripture_chap_box
1413       {
1414         \box_move_down:nn
1415           { \baselineskip }
1416           {
1417             \hbox_overlap_left:n
1418               {
1419                 \box_use:N \l__scripture_chap_tmp_box
1420                 \skip_horizontal:N \l__scripture_chap_sep_tl
1421               }
1422           }
1423       }
1424     \box_set_ht:Nn \l__scripture_chap_box { \c_zero_dim }
1425     \box_set_dp:Nn \l__scripture_chap_box { \c_zero_dim }
1426     \bool_if:NTF \l__scripture_mid_para_chap_bool
1427       { \__scripture_mid_para_drop_chap_output: }
1428       {
```

```
1429            \para_end:
1430            \__scripture_drop_chap_output:
1431            \str_if_eq:VnF \l__scripture_currenvir_str { midparachap }
1432              { \__scripture_pilcrow_output: }
1433          }
1434       \hook_gput_next_code:nn { para / before }
1435         {
1436           \legacy_if:nF { @newlist }
1437             {
1438               \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
1439                 {
1440                   \dim_compare:nNnT \parskip < \baselineskip
1441                     {
1442                       \dim_compare:nNnT { \g__scripture_chap_width_dim + \l__scripture_chap_se
1443                         {
1444                           \hook_gput_next_code:nn { para / begin }
1445                             {
1446                               \para_omit_indent:
1447                               \skip_horizontal:n {\g__scripture_chap_width_dim + \l__scripture
1448                             }
1449                         }
1450                     }
1451                 }
1452               \int_gzero:N \g__scripture_chap_par_prevgraf_int
1453             }
1454           \parshape 1 ~ \@totalleftmargin ~ \linewidth
1455         }
1456     }
```

(*End of definition for* `\__scripture_chap:n`.)

3.5.5 Formatting and printing verse numbers

`\l_scripture_verse_par_start_bool` Flag for whether the current verse is at the start of a paragraph.

```
1457 \bool_new:N \l__scripture_verse_par_start_bool
```

(*End of definition for* `\l__scripture_verse_par_start_bool`.)

`\l_scripture_no_verse_first_bool` Suppress the next verse from formatting differently if it begins a new paragraph.

```
1458 \bool_new:N \l__scripture_no_verse_first_bool
```

(*End of definition for* `\l__scripture_no_verse_first_bool`.)

`\g_scripture_suppress_next_verse_para_bool` Suppress the next verse from starting a new paragraph when `verse/para` is in use.

```
1459 \bool_new:N \g__scripture_suppress_next_verse_para_bool
```

(*End of definition for* `\g__scripture_suppress_next_verse_para_bool`.)

`\__scripture_verse_output:n` Output a formatted verse.

```
1460 \cs_new_protected:Nn \__scripture_verse_output:n
1461   {
1462     \cs_gset_nopar:Npn \scripturecurrentverse { #1 }
1463     \legacy_if:nTF { @endpe }
1464       { \bool_set_false:N \l__scripture_verse_par_start_bool }
1465       { \bool_set_true:N \l__scripture_verse_par_start_bool }
```

```
1466    \mode_if_vertical:TF
1467      { \mode_leave_vertical: }
1468      { \bool_set_false:N \l__scripture_verse_par_start_bool }
1469    \bool_if:NT \l__scripture_no_verse_first_bool
1470      {
1471        \bool_set_false:N \l__scripture_verse_par_start_bool
1472        \bool_set_false:N \l__scripture_no_verse_first_bool
1473      }
1474    \group_begin:
1475    \bool_if:NT \l__scripture_verse_para_bool
1476      {
1477        \box_if_empty:NF \g__scripture_post_env_indent_box
1478          { \box_use_drop:N \g__scripture_post_env_indent_box }
1479      }
1480    \hook_use:n { scripture / verse / before }
1481    \bool_if:NT \l__scripture_verse_show_bool
1482      {
1483        \group_begin:
1484        \l__scripture_verse_font_tl
1485        \str_if_eq:VnF \l__scripture_verse_colour_tl { . }
1486          { \exp_args:NV \color_select:n \l__scripture_verse_colour_tl }
1487        \bool_lazy_all:nTF
1488          {
1489            { \l__scripture_verse_first_bool }
1490            { \l__scripture_verse_par_start_bool }
1491            { \bool_not_p:n { \str_if_eq_p:Vn \l__scripture_currenvir_str { poetry } } } }
1492          }
1493          {
1494            \__scripture_verse_first_format:n { #1 }
1495            \group_end:
1496            \kern \l__scripture_verse_first_sep_tl
1497          }
1498          {
1499            \__scripture_verse_format:n { #1 }
1500            \group_end:
1501            \kern \l__scripture_verse_sep_tl
1502          }
1503        \nobreak
1504        \skip_horizontal:N \c_zero_skip
1505      }
1506    \hook_use:n { scripture / verse / after }
1507    \hook_use:n { scripture / pilcrow }
1508    \group_end:
1509  }
```

(*End of definition for* `\__scripture_verse_output:n`.)

`\__scripture_verse_compact_output:n`   Output a formatted verse.

```
1510 \cs_new_protected:Nn \__scripture_verse_compact_output:n
1511   {
1512     \mode_if_vertical:T
1513       { \__scripture_set_pilcrow_hook: }
1514     \__scripture_verse_output:n { #1 }
1515   }
```

(End of definition for __scripture_verse_compact_output:n.)

__scripture_verse_para_output:n Output a formatted verse when verse/para is set.

```
1516 \cs_new_protected:Nn \__scripture_verse_para_output:n
1517   {
1518     \mode_if_vertical:T
1519       { \__scripture_set_pilcrow_hook: }
1520     \hook_if_empty:nT { scripture / pilcrow }
1521       { \dim_set:Nn \parindent \l__scripture_para_indent_tl }
1522     \bool_if:NTF \g__scripture_suppress_next_verse_para_bool
1523       { \bool_gset_false:N \g__scripture_suppress_next_verse_para_bool }
1524       {
1525         \legacy_if:nF { @newlist }
1526           {
1527             \mode_if_vertical:F
1528               {
1529                 \para_end:
1530                 \parshape 1 ~ \@totalleftmargin ~ \linewidth
1531               }
1532           }
1533       }
1534     \__scripture_verse_output:n { #1 }
1535   }
```

(End of definition for __scripture_verse_para_output:n.)

__scripture_hanging_verse_para:n Output a formatted verse when verse/para is set and in a hanging environment.

```
1536 \cs_new_protected:Nn \__scripture_hanging_verse_para:n
1537   {
1538     \mode_if_vertical:T
1539       { \__scripture_set_pilcrow_hook: }
1540     \bool_if:NF \g__scripture_suppress_next_verse_para_bool
1541       {
1542         \legacy_if:nF { @newlist }
1543           {
1544             \mode_if_vertical:F
1545               {
1546                 \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
1547                 \hook_gput_next_code:nn { para / begin }
1548                   { \parshape 1 ~ \@totalleftmargin ~ \linewidth }
1549                 \par
1550                 \tex_noindent:D
1551               }
1552           }
1553       }
1554     \bool_gset_false:N \g__scripture_suppress_next_verse_para_bool
1555     \__scripture_hanging_verse:n { #1 }
1556   }
```

(End of definition for __scripture_hanging_verse_para:n.)

3.5.6 Printing pilcrows

\g__scripture_suppress_next_pilcrow_bool   Flag to suppress the next pilcrow.

```
1557 \bool_new:N \g__scripture_suppress_next_pilcrow_bool
```

(*End of definition for* \g__scripture_suppress_next_pilcrow_bool.)

__scripture_set_pilcrow_hook: Set the next `scripture/pilcrow` hook to output a pilcrow providing the hook is currently empty. `verse/para` or `compact` options are set.

```
1558 \cs_new_protected:Nn \__scripture_set_pilcrow_hook:
1559   {
1560     \hook_if_empty:nT { scripture / pilcrow }
1561       {
1562         \hook_gput_next_code:nn { scripture / pilcrow } { \__scripture_pilcrow_output: }
1563       }
1564   }
```

(*End of definition for* __scripture_set_pilcrow_hook:.)

__scripture_pilcrow_output: Output a pilcrow (¶). Pilrows only display when either the `verse/para` or `compact` options are set.

```
1565 \cs_new_protected:Nn \__scripture_pilcrow_output:
1566   {
1567     \bool_gset_false:N \g__scripture_para_mode_vertical_bool
1568     \bool_gset_false:N \g__scripture_mode_vertical_inner_bool
1569     \bool_if:NT \l__scripture_pilcrow_bool
1570       {
1571         \bool_lazy_or:nnT \l__scripture_verse_para_bool \l__scripture_compact_bool
1572           {
1573             \bool_if:NTF \g__scripture_suppress_next_pilcrow_bool
1574               { \bool_gset_false:N \g__scripture_suppress_next_pilcrow_bool }
1575               {
1576                 \P
1577                 \kern \l__scripture_pilcrow_sep_tl
1578                 \nobreak
1579                 \skip_horizontal:N \c_zero_skip
1580               }
1581           }
1582       }
1583   }
```

(*End of definition for* __scripture_pilcrow_output:.)

3.5.7 Calculating the length of the final line in a paragraph

\g__scripture_final_line_dim   Store the length of the final line in a paragraph.

```
1584 \dim_new:N \g__scripture_final_line_dim
```

(*End of definition for* \g__scripture_final_line_dim.)

__scripture_calc_final_line_length: Calculate the length of the final line in a paragraph.

```
1585 \cs_new_protected:Nn \__scripture_calc_final_line_length:
1586   {
1587     \skip_set:Nn \abovedisplayshortskip { -\baselineskip }
1588     \skip_set_eq:NN \abovedisplayskip \abovedisplayshortskip
```

```
1589        \skip_zero:N \belowdisplayshortskip
1590        \skip_zero:N \belowdisplayskip
1591      $$
1592        \hook_use:n { para / after }
1593        \dim_compare:nNnTF \predisplaysize > \c_zero_dim
1594          {
1595            \dim_gset:Nn \g__scripture_final_line_dim
1596              { \predisplaysize - 2em }
1597          }
1598          {
1599            \bool_if:NT \l__scripture_textdir_change_in_group_bool
1600              {
1601                \str_if_eq:VnT \l__scripture_currenvir_str { scripture }
1602                  { \int_set:Nn \prevgraf { \int_use:N \prevgraf - 1 } }
1603              }
1604          }
1605        \cs_gset_nopar:Nx \__scripture_reset_spacing:
1606          {
1607            \dim_set:Nn \predisplaysize { \dim_use:N \predisplaysize }
1608            \int_set:Nn \prevgraf { \int_use:N \prevgraf }
1609            \int_gset:Nn \g__scripture_chap_par_prevgraf_int { \int_use:N \g__scripture_chap_p
1610          }
1611      $$
1612      \__scripture_reset_spacing:
1613      \dim_compare:nNnT \g__scripture_final_line_dim > \paperwidth
1614        { \dim_gset:Nn \g__scripture_final_line_dim { \@totalleftmargin + \linewidth } }
1615      \dim_compare:nNnT \g__scripture_final_line_dim > { \c_zero_dim - 1sp }
1616        { \skip_vertical:n { -\baselineskip - \parskip } }
1617    }
```

(*End of definition for* `\__scripture_calc_final_line_length:`.)

3.5.8   The `scripture` environment

`\l__scripture_currenvir_str`   Store the current `scripture` environment.

```
1618 \str_new:N \l__scripture_currenvir_str
```

(*End of definition for* `\l__scripture_currenvir_str`.)

`\l_scripture_parskip_correction_skip`   Correction to ensure consistent `\parskip` at start of `scripture` environment with non-zero `parskip` option.

```
1619 \skip_new:N \l__scripture_parskip_correction_skip
```

(*End of definition for* `\l__scripture_parskip_correction_skip`.)

`\l__scripture_ref_tl`   Store the reference passed to the `scripture` environment.

```
1620 \tl_new:N \l__scripture_ref_tl
```

(*End of definition for* `\l__scripture_ref_tl`.)

`\__scripture_added:n`   Mark text as added.

```
1621 \cs_new_protected_nopar:Nn \__scripture_added:n
1622   {
1623     \group_begin:
1624     \l__scripture_added_font_tl
```

66

```
1625        \__scripture_added_format:n { #1 }
1626      \group_end:
1627    }
```

(*End of definition for* __scripture_added:n.)

__scripture_heading:n Output a non-canonical heading.

```
1628 \cs_new_protected_nopar:Nn \__scripture_heading:n
1629   {
1630     \str_if_eq:VnTF \l__scripture_currenvir_str { scripture }
1631       {
1632         \bool_if:NTF \l__scripture_heading_show_bool
1633           {
1634             \parshape 1 ~ \@totalleftmargin ~ \linewidth
1635             \group_begin:
1636             \bool_set_false:N \l__scripture_pilcrow_bool
1637             \legacy_if:nTF { @newlist }
1638               {
1639                 \skip_horizontal:n { -\parindent }
1640               }
1641               {
1642                 \__scripture_list_para_end:
1643                 \addvspace { \l__scripture_heading_aboveskip_tl }
1644                 \tex_noindent:D
1645               }
1646             \l__scripture_heading_align_tl
1647             \l__scripture_heading_font_tl
1648             \__scripture_heading_format:n { #1 }
1649             \__scripture_list_para_end:
1650             \group_end:
1651             \bool_if:NTF \l__scripture_verse_para_bool
1652               {
1653                 \cs_set_eq:NN \noindent \__scripture_para_noindent:
1654               }
1655               {
1656                 \cs_set_eq:NN \noindent \tex_noindent:D
1657               }
1658             \legacy_if_set:nn { @afterindent } \l__scripture_heading_afterindent_bool
1659             \bool_if:NF \l__scripture_heading_afterindent_bool
1660               {
1661                 \hook_gput_next_code:nn { para / before }
1662                   { \para_omit_indent: }
1663               }
1664             \nobreak
1665             \@afterheading
1666             \addvspace { \l__scripture_heading_belowskip_tl }
1667           }
1668           { \ignorespaces }
1669       }
1670       {
1671         \msg_error:nnx { scripture } { heading-error } { \l__scripture_currenvir_str }
1672       }
1673   }
```

(*End of definition for* __scripture_heading:n.)

`\__scripture_name:n` Output divine name.

```
1674 \cs_new_protected_nopar:Nn \__scripture_name:n
1675   {
1676     \group_begin:
1677     \l__scripture_name_font_tl
1678     \__scripture_name_format:n { #1 }
1679     \group_end:
1680   }
```

(*End of definition for* `\__scripture_name:n`.)

`\__scripture_no_first_verse:` Suppress special formatting for first verse if active.

```
1681 \cs_new_protected_nopar:Nn \__scripture_no_first_verse:
1682   {
1683     \bool_set_true:N \l__scripture_no_verse_first_bool
1684   }
```

(*End of definition for* `\__scripture_no_first_verse:`.)

`\__scripture_ch_output:nn` Print a chapter number in the main scripture environment.

```
1685 \cs_new_protected:Nn \__scripture_ch_output:nn
1686   {
1687     \bool_if:NTF \l__scripture_chap_show_bool
1688       {
1689         \bool_if:NTF \l__scripture_chap_drop_bool
1690           {
1691             \__scripture_chap:n { #2 }
1692             \bool_lazy_and:nnTF
1693             { #1 }
1694             { \bool_not_p:n { \l__scripture_compact_bool } }
1695             {
1696               \hook_gput_next_code:nn { para / after }
1697                 {
1698                   \int_gzero:N \g__scripture_chap_par_prevgraf_int
1699                   \int_gzero:N \g__scripture_mid_para_chap_prevgraf_int
1700                 }
1701             }
1702             {
1703               \hook_gput_next_code:nn { para / after }
1704                 { \int_gset_eq:NN \g__scripture_mid_para_chap_prevgraf_int \prevgraf }
1705             }
1706           }
1707           {
1708             \bool_if:NTF \l__scripture_chap_para_bool
1709               { \__scripture_para_chap:n { #2 } }
1710               { \__scripture_nodrop_chap:n { #2 } }
1711           }
1712       }
1713       {
1714         \bool_if:NTF \l__scripture_chap_para_bool
1715           { \__scripture_para_chap:n { #2 } }
1716           { \__scripture_nodrop_chap:n { #2 } }
1717       }
1718   }
```

68

(End of definition for `\__scripture_ch_output:nn`.*)*

`\__scripture_begin:nn`   Begin a scripture quotation.

```
1719 \cs_new_protected:Nn \__scripture_begin:nn
1720   {
1721     \cs_set_eq:NN \__scripture_noindent_saved: \noindent
1722     \bool_if:NT \l__scripture_active_bool
1723       {
1724         \msg_error:nn { scripture } { nested-environment }
1725       }
1726     \bool_set_true:N \l__scripture_active_bool
1727     \tl_if_novalue:nF { #1 }
1728       { \tl_set:Nn \l__scripture_ref_tl { #1 } }
1729     \str_set:Nn \l__scripture_currenvir_str { scripture }
1730     \tl_if_novalue:nF { #2 }
1731       {
1732         \keys_set:nn { scripture } { #2 }
1733       }
1734     \l__scripture_before_tl
1735     \__scripture_select_language:
1736     \int_gzero:N \g__scripture_chap_par_prevgraf_int
```

Empty `\everydisplay` to ensure it isn't interring with spacing.

```
1737     \everydisplay { }
1738     \int_set:Nn \predisplaypenalty { 10000 }
1739     \int_set_eq:NN \postdisplaypenalty \@lowpenalty
1740     \int_set:Nn \@beginparpenalty { -\@lowpenalty }
1741     \int_set_eq:NN \@endparpenalty \@lowpenalty
1742     \bool_gset_false:N \g__scripture_poetry_end_bool
1743     \cs_set_eq:NN \extraskip \__scripture_extra_skip:
1744     \cs_set_eq:NN \redletteron \__scripture_red_letter_on:
1745     \cs_set_eq:NN \redletteroff \__scripture_red_letter_off:
1746     \cs_set:Npn \nopilcrow { \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool }
1747     \bool_gset_false:N \g__scripture_suppress_next_pilcrow_bool
1748     \bool_gset_false:N \g__scripture_suppress_next_verse_para_bool
1749     \bool_gset_false:N \g__scripture_mode_vertical_inner_bool
1750     \box_gclear:N \g__scripture_post_env_indent_box
1751     \skip_gzero:N \g__scripture_prev_inner_below_skip
1752     \bool_if:NT \l__scripture_chap_para_bool
1753       { \bool_set_false:N \l__scripture_chap_drop_bool }
1754     \DeclareDocumentCommand { \ch } { s o m }
1755       {
1756         \tl_if_novalue:nF { ##2 }
1757           {
1758             \tl_set:Nn \l__scripture_X_char_tl { ##2 }
1759           }
1760         \__scripture_ch_output:nn { ##1 } { ##3 }
1761       }
1762     \cs_set_eq:NN \vs \__scripture_verse_output:n
1763     \cs_set_eq:NN \added \__scripture_added:n
1764     \cs_set_eq:NN \heading \__scripture_heading:n
1765     \cs_set_eq:NN \name \__scripture_name:n
1766     \cs_set_protected_nopar:Npn \LORD { \name { Lord } }
1767     \cs_set_protected_nopar:Npn \GOD { \name { God } }
```

69

```
1768        \cs_set_eq:NN \nofirstverse \__scripture_no_first_verse:
1769        \cs_set_protected_nopar:Npn \textright ##1
1770          { \__scripture_text_right:nn { \l__scripture_text_right_sep_tl } { ##1 } }
1771        \hook_gclear_next_code:n { para / before }
1772        \hook_gclear_next_code:n { scripture / pilcow }
1773        \hook_gclear_next_code:n { scripture / poetry / pilcow }
1774        \l__scripture_font_tl
1775        \tl_set:Ne \l__scripture_parindent_tl { \dim_eval:n { \l__scripture_parindent_tl } }
1776        \exp_args:NnV \color_set:nn { scripture default colour } \l__scripture_colour_tl
1777        \str_if_eq:VnF \l__scripture_colour_tl { . }
1778          { \color_select:n { scripture default colour } }
1779        \bool_if:NTF \l__scripture_inline_bool
1780          { \l__scripture_inline_begin_tl }
1781          {
1782            \legacy_if:nTF { @newlist }
1783              { \dim_set_eq:NN \l__scripture_outer_itemindent_dim \itemindent }
1784              { \dim_zero:N \l__scripture_outer_itemindent_dim }
1785            \list { }
1786              {
1787                \dim_set_eq:NN \leftmargin \l__scripture_leftmargin_tl
1788                \dim_set_eq:NN \rightmargin \l__scripture_rightmargin_tl
1789                \dim_set:Nn \itemindent { \l__scripture_parindent_tl - \l__scripture_outer_itemi
1790                \dim_set_eq:NN \listparindent \l__scripture_parindent_tl
1791                \skip_zero:N \partopsep
1792                \skip_set_eq:NN \parsep \l__scripture_parskip_tl
1793                \dim_compare:nNnTF \parskip > \l__scripture_aboveskip_tl
1794                  { \skip_zero:N \topsep }
1795                  { \skip_set:Nn \topsep { \__scripture_skip_diff:NN \l__scripture_aboveskip_tl
1796                \__scripture_setup_list_noindent:n { \l__scripture_parindent_tl }
1797                \bool_if:NF \l__scripture_indent_bool
1798                  {
1799                    \hook_gput_next_code:nn { para / begin }
1800                      {
1801                        \bool_if:NTF \l__scripture_compact_bool
1802                          { \para_omit_indent: }
1803                          {
1804                            \bool_lazy_any:nT
1805                              {
1806                                { \str_if_eq_p:Vn \l__scripture_currenvir_str { midparachap } }
1807                                { \str_if_eq_p:Vn \l__scripture_currenvir_str { narrow } }
1808                                { \str_if_eq_p:Vn \l__scripture_currenvir_str { scripture } }
1809                              }
1810                              {
1811                                \legacy_if:nTF { @noparlist }
1812                                  {
1813                                    \bool_lazy_and:nnF
1814                                      { \str_if_eq_p:Vn \l__scripture_currenvir_str { narrow } }
1815                                      { \str_if_eq_p:Vn \l__scripture_ref_position_tl { start }
1816                                      { \noindent }
1817                                  }
1818                                  { \para_omit_indent: }
1819                              }
1820                          }
1821                      }
```

70

```
1822                    }
1823                }
1824            \cs_set_eq:NN \__scripture_list_para_end: \@par
1825            \item
1826            \relax
1827        }
1828    }
```

(*End of definition for* __scripture_begin:nn.)

__scripture_end: End a scripture quotation with a reference.

```
1829 \cs_new_protected:Nn \__scripture_end:
1830    {
1831        \tl_if_eq:NnT \l__scripture_ref_position_tl { start }
1832            {
1833                \tl_clear:N \l__scripture_ref_tl
1834                \tl_if_eq:NnT \l__scripture_version_position_tl { withref }
1835                    { \tl_clear:N \l__scripture_version_tl }
1836            }
1837        \cs_set_eq:NN \noindent \tex_noindent:D
1838        \bool_if:NTF \l__scripture_inline_bool
1839            {
1840                \unskip
1841                \__scripture_red_letter_off:
1842                \l__scripture_inline_end_tl
1843                \bool_lazy_or:nnTF
1844                    { \tl_if_novalue_p:n \l__scripture_ref_tl }
1845                    { \tl_if_empty_p:N \l__scripture_ref_tl }
1846                    {
1847                        \tl_if_empty:NF \l__scripture_version_tl
1848                            {
1849                                \skip_horizontal:N \l__scripture_ref_sep_tl
1850                                \__scripture_format_full_ref:nn { } \l__scripture_version_tl
1851                            }
1852                    }
1853                    {
1854                        \skip_horizontal:N \l__scripture_ref_sep_tl
1855                        \__scripture_format_full_ref:nn \l__scripture_ref_tl \l__scripture_version_tl
1856                    }
1857            }
1858            {
1859                \__scripture_red_letter_off:
1860                \skip_zero:N \parskip
1861                \int_set:Nn \postdisplaypenalty { 10000 }
1862                \__scripture_calc_final_line_length:
1863                \dim_gsub:Nn \g__scripture_final_line_dim \@totalleftmargin
1864                \dim_compare:nNnTF \@outerparskip > \l__scripture_belowskip_tl
1865                    { \skip_zero:N \@topsepadd }
1866                    { \skip_set:Nn \@topsepadd { \__scripture_skip_diff:NN \l__scripture_belowskip_tl
```

If a `scripture` quotation ends with an inner environment, remove the below skip of the
inner environment. Also correct for \textdir quirk.

```
1867                \int_compare:nNnT \prevgraf = \c_zero_int
1868                    {
1869                        \bool_if:NT \l__scripture_textdir_change_in_group_bool
```

71

```
1870                { \skip_vertical:n { -\baselineskip } }
1871              \skip_vertical:n { -\g__scripture_prev_inner_below_skip }
1872            }
1873          \hook_gclear_next_code:n { para / before }
1874          \noindent
1875          \bool_lazy_or:nnTF
1876            { \tl_if_novalue_p:n \l__scripture_ref_tl }
1877            { \tl_if_empty_p:N \l__scripture_ref_tl }
1878            {
1879              \tl_if_empty:NTF \l__scripture_version_tl
1880                { \hbox:n { } }
1881                { \__scripture_reference:n { } }
1882            }
1883            { \__scripture_reference:n \l__scripture_ref_tl }
1884          \parshape 1 ~ \@totalleftmargin ~ \linewidth
```

Set @noparlist to false to ensure \topsep is always added scripture environment

```
1885          \legacy_if_set_false:n { @noparlist }
1886          \endlist
1887        }
1888      \cs_set_eq:NN \noindent \__scripture_noindent_saved:
1889      \l__scripture_after_tl
1890    }
```

(*End of definition for* __scripture_end:.)

3.6   Scripture internal environments

__scripture_compact_end_inner: End a compact inner environment

```
1891 \cs_new_protected:Nn \__scripture_compact_end_inner:
1892   {
1893     \bool_gset_false:N \g__scripture_mode_vertical_inner_bool
1894     \str_if_eq:VnT \l__scripture_ref_position_tl { start }
1895       {
1896         \tl_gclear:N \l__scripture_ref_tl
1897       }
1898     \bool_lazy_and:nnT \l__scripture_verse_para_bool \l__scripture_chap_drop_bool
1899       {
1900         \cs_set_eq:NN \par \para_end:
1901         \__scripture_calc_final_line_length:
1902         \int_gset_eq:NN \g__scripture_mid_para_chap_prevgraf_int \g__scripture_chap_par_prev
1903         \tex_noindent:D
1904         \bool_lazy_and:nnTF
1905           { \bool_if_p:N \l__scripture_compact_bool }
1906           { \int_compare_p:nNn \g__scripture_mid_para_chap_prevgraf_int = 1 }
1907           {
1908             \hook_gput_next_code:nn { env / \l__scripture_currenvir_str / after }
1909               {
1910                 \parshape 3 ~
1911                   \@totalleftmargin ~ \linewidth ~
1912                   \dim_eval:n
1913                     { \@totalleftmargin + \g__scripture_chap_width_dim + \l__scripture_chap_
1914                   \dim_eval:n
1915                     { \linewidth - \g__scripture_chap_width_dim - \l__scripture_chap_sep_tl
```

72

```
1916                    \@totalleftmargin ~ \linewidth
1917                  }
1918                \hbox_to_wd:nn { \g__scripture_final_line_dim - \g__scripture_chap_width_dim - \
1919              }
1920              {
1921                \hbox_to_wd:nn \g__scripture_final_line_dim { }
1922              }
1923          \c_space_token
1924          \kern 0pt
1925          \cs_set_eq:NN \par \relax
1926          \hook_gput_next_code:nn { para / before }
1927            {
1928              \int_compare:nNnT \g__scripture_mid_para_chap_prevgraf_int = 1
1929                {
1930                  \dim_compare:nNnT \parskip < \baselineskip
1931                    {
1932                      \dim_compare:nNnT { \g__scripture_chap_width_dim + \l__scripture_chap_se
1933                        {
1934                          \hook_gput_next_code:nn { para / begin }
1935                            {
1936                              \para_omit_indent:
1937                              \skip_horizontal:n {\g__scripture_chap_width_dim + \l__scripture
1938                            }
1939                        }
1940                    }
1941                }
1942              \int_gzero:N \g__scripture_chap_par_prevgraf_int
1943              \parshape 1 ~ \@totalleftmargin ~ \linewidth
1944            }
1945        }
1946      \unskip
1947    }
```

(*End of definition for* `\__scripture_compact_end_inner:`.)

3.6.1   The `center` environment

`\__scripture_center_begin:n`   Begin a centred section in a scripture quotation.

```
1948 \cs_new_protected:Nn \__scripture_center_begin:n
1949   {
1950     \bool_if:NT \l__scripture_active_inner_bool
1951       {
1952         \msg_error:nn { scripture } { nested-environment }
1953       }
1954     \bool_set_true:N \l__scripture_active_inner_bool
1955     \str_set:Nn \l__scripture_currenvir_str { center }
1956     \tl_if_novalue:nF { #1 }
1957       {
1958         \keys_set:nn
1959           { scripture / center }
1960           { #1 }
1961       }
1962     \legacy_if:nTF { @newlist }
1963       { \dim_add:Nn \l__scripture_outer_itemindent_dim \itemindent }
```

```
1964            { \dim_zero:N \l__scripture_outer_itemindent_dim }
1965        \cs_set_eq:NN \par \para_end:
1966        \list { }
1967          {
1968            \bool_lazy_or:nnF
1969              { \mode_if_vertical_p: }
1970              { \g__scripture_para_mode_vertical_bool }
1971              { \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool }
1972            \dim_zero:N \labelsep
1973            \dim_set_eq:NN \leftmargin \l__scripture_center_leftmargin_tl
1974            \dim_set_eq:NN \rightmargin \l__scripture_center_rightmargin_tl
1975            \dim_set:Nn \itemindent { -\l__scripture_outer_itemindent_dim }
1976            \dim_zero:N \listparindent
1977            \skip_set_eq:NN \parsep \parskip
1978            \skip_zero:N \partopsep
1979            \dim_compare:nNnTF \parskip > \l__scripture_center_aboveskip_tl
1980              { \skip_zero:N \topsep }
1981              { \skip_set:Nn \topsep { \__scripture_skip_diff:NN \l__scripture_center_aboveskip_
1982            \__scripture_setup_list_noindent:n { \c_zero_dim }
1983          }
1984        \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
1985          { \addvspace { \baselineskip } }
1986        \int_gzero:N \g__scripture_chap_par_prevgraf_int
1987        \centering
1988        \bool_if:NT \l__scripture_verse_para_bool
1989          {
1990            \cs_set_eq:NN \par \__scripture_para_par:
1991            \cs_set_protected:Npn \\
1992              {
1993                \mode_if_horizontal:TF
1994                  { \unskip }
1995                  { \@nolnerr }
1996              }
1997          }
1998        \item
1999        \relax
2000      }
```

(*End of definition for* __scripture_center_begin:n.)

__scripture_center_end: End a centred section in a scripture quotation.

```
2001 \cs_new_protected:Nn \__scripture_center_end:
2002   {
2003     \cs_set_eq:NN \par \para_end:
2004     \dim_gset:Nn \g__scripture_final_line_dim { \@totalleftmargin + \linewidth }
2005     \dim_compare:nNnTF \@outerparskip > \l__scripture_center_belowskip_tl
2006       { \skip_zero:N \@topsepadd }
2007       { \skip_set:Nn \@topsepadd { \__scripture_skip_diff:NN \l__scripture_center_belowskip_
2008     \skip_gset_eq:NN \g__scripture_prev_inner_below_skip \@topsepadd
2009     \legacy_if_set_false:n { @noparlist }
2010     \endlist
2011     \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
2012   }
```

(*End of definition for* __scripture_center_end:.)

3.6.2 The `flushright` environment

`\__scripture_flushright_begin:n` Begin a flushright section in a scripture quotation.

```
2013 \cs_new_protected:Nn \__scripture_flushright_begin:n
2014   {
2015     \bool_if:NT \l__scripture_active_inner_bool
2016       {
2017         \msg_error:nn { scripture } { nested-environment }
2018       }
2019     \bool_set_true:N \l__scripture_active_inner_bool
2020     \str_set:Nn \l__scripture_currenvir_str { flushright }
2021     \tl_if_novalue:nF { #1 }
2022       {
2023         \keys_set:nn
2024           { scripture / flushright }
2025           { #1 }
2026       }
2027     \legacy_if:nTF { @newlist }
2028       { \dim_add:Nn \l__scripture_outer_itemindent_dim \itemindent }
2029       { \dim_zero:N \l__scripture_outer_itemindent_dim }
2030     \cs_set_eq:NN \par \para_end:
2031     \list { }
2032       {
2033         \bool_lazy_or:nnF
2034           { \mode_if_vertical_p: }
2035           { \g__scripture_para_mode_vertical_bool }
2036           { \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool }
2037         \dim_zero:N \labelsep
2038         \dim_set_eq:NN \leftmargin \l__scripture_flushright_leftmargin_tl
2039         \dim_set_eq:NN \rightmargin \l__scripture_flushright_rightmargin_tl
2040         \dim_set:Nn \itemindent { -\l__scripture_outer_itemindent_dim }
2041         \dim_zero:N \listparindent
2042         \skip_set_eq:NN \parsep \parskip
2043         \skip_zero:N \partopsep
2044         \dim_compare:nNnTF \parskip > \l__scripture_flushright_aboveskip_tl
2045           { \skip_zero:N \topsep }
2046           { \skip_set:Nn \topsep { \__scripture_skip_diff:NN \l__scripture_flushright_aboves
2047         \__scripture_setup_list_noindent:n { \c_zero_dim }
2048       }
2049     \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
2050       { \addvspace { \baselineskip } }
2051     \int_gzero:N \g__scripture_chap_par_prevgraf_int
2052     \raggedleft
2053     \bool_if:NT \l__scripture_verse_para_bool
2054       {
2055         \cs_set_eq:NN \par \__scripture_para_par:
2056         \cs_set_protected:Npn \\
2057           {
2058             \mode_if_horizontal:TF
2059               { \unskip }
2060               { \@nolnerr }
2061           }
2062       }
2063     \item
```

```
2064        \relax
2065     }
```

(*End of definition for* `\__scripture_flushright_begin:n.`)

`\__scripture_flushright_end:` End a flushright section in a scripture quotation.

```
2066 \cs_new_protected:Nn \__scripture_flushright_end:
2067   {
2068     \cs_set_eq:NN \par \para_end:
2069     \dim_gset:Nn \g__scripture_final_line_dim { \@totalleftmargin + \linewidth }
2070     \dim_compare:nNnTF \@outerparskip > \l__scripture_flushright_belowskip_tl
2071       { \skip_zero:N \@topsepadd }
2072       { \skip_set:Nn \@topsepadd { \__scripture_skip_diff:NN \l__scripture_flushright_belows
2073     \skip_gset_eq:NN \g__scripture_prev_inner_below_skip \@topsepadd
2074     \legacy_if_set_false:n { @noparlist }
2075     \endlist
2076     \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
2077   }
```

(*End of definition for* `\__scripture_flushright_end:.`)

3.6.3   The hanging environment

`\l_scripture_hanging_parindent_saved_dim` Save the `\parindent` for later use in hanging environment.

```
2078 \dim_new:N \l__scripture_hanging_parindent_saved_dim
```

(*End of definition for* `\l__scripture_hanging_parindent_saved_dim.`)

`\__scripture_hanging_ch:n` Position a chapter in a hanging environment.

```
2079 \cs_new_protected:Nn \__scripture_hanging_ch:n
2080   {
2081     \bool_gset_false:N \g__scripture_suppress_next_pilcrow_bool
2082     \bool_if:NTF \l__scripture_chap_show_bool
2083       {
2084         \bool_if:NTF \l__scripture_chap_drop_bool
2085           { \__scripture_hanging_chap:n { #1 } }
2086           {
2087             \bool_if:NTF \l__scripture_chap_para_bool
2088               { \__scripture_para_chap:n { #1 } }
2089               { \__scripture_nodrop_chap:n { #1 } }
2090           }
2091       }
2092       {
2093         \bool_if:NTF \l__scripture_chap_para_bool
2094           { \__scripture_para_chap:n { #1 } }
2095           { \__scripture_nodrop_chap:n { #1 } }
2096       }
2097   }
```

(*End of definition for* `\__scripture_hanging_ch:n.`)

`\__scripture_hanging_chap:n` Position a drop chapter in a hanging environment.

```
2098 \cs_new_protected:Nn \__scripture_hanging_chap:n
2099   {
2100     \__scripture_drop_chap_set_up:n { #1 }
```

```
2101      \dim_compare:nNnTF
2102        { \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl } < \leftmargin
2103        {
2104          \bool_if:NT \l__scripture_hanging_chapter_indent_bool
2105            {
2106              \tl_set:Nx \l__scripture_chap_sep_tl
2107                {
2108                  \dim_eval:n { \leftmargin - \g__scripture_chap_width_dim }
2109                }
2110            }
2111        }
2112        {
2113          \parshape 3 ~
2114            \dim_eval:n
2115              {
2116                \@totalleftmargin - \leftmargin + \g__scripture_chap_width_dim +
2117                \l__scripture_chap_sep_tl
2118              } ~
2119            \dim_eval:n
2120              {
2121                \linewidth + \leftmargin - \g__scripture_chap_width_dim -
2122                \l__scripture_chap_sep_tl
2123              } ~
2124            \dim_eval:n
2125              {
2126                \@totalleftmargin - \leftmargin + \g__scripture_chap_width_dim +
2127                \l__scripture_chap_sep_tl
2128              } ~
2129            \dim_eval:n
2130              {
2131                \linewidth + \leftmargin - \g__scripture_chap_width_dim -
2132                \l__scripture_chap_sep_tl
2133              } ~
2134          \@totalleftmargin ~ \linewidth
2135        }
2136      \par
2137      \bool_if:NTF \l__scripture_hanging_chapter_indent_bool
2138        {
2139          \tl_set:Nx \l__scripture_chap_sep_tl
2140            {
2141              \dim_eval:n
2142                {
2143                  \l__scripture_chap_sep_tl + \l__scripture_hanging_parindent_saved_dim
2144                }
2145            }
2146          \__scripture_nohang:
2147        }
2148        {
2149          \mode_leave_vertical:
2150          \dim_compare:nNnTF
2151            { \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl } > \l__scripture_hangi
2152            {
2153              \dim_compare:nNnTF
2154                { \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl } > \leftmargin
```

```
2155            {
2156              \skip_horizontal:n { \l__scripture_hanging_hang_tl }
2157            }
2158            {
2159              \skip_horizontal:n
2160                {
2161                  \l__scripture_chap_sep_tl +
2162                  \g__scripture_chap_width_dim -
2163                  \l__scripture_hanging_leftmargin_tl
2164                }
2165            }
2166          }
2167          {
2168            \tl_set:Nx \l__scripture_chap_sep_tl
2169              {
2170                \dim_eval:n
2171                  { \l__scripture_hanging_leftmargin_tl - \g__scripture_chap_width_dim }
2172              }
2173          }
2174      }
2175    \hbox_set:Nn \l__scripture_chap_box
2176      {
2177        \box_move_down:nn
2178          { \baselineskip }
2179          {
2180            \hbox_overlap_left:n
2181              {
2182                \box_use:N \l__scripture_chap_tmp_box
2183                \skip_horizontal:N \l__scripture_chap_sep_tl
2184              }
2185          }
2186      }
2187    \box_set_ht:Nn \l__scripture_chap_box { \c_zero_dim }
2188    \box_set_dp:Nn \l__scripture_chap_box { \c_zero_dim }
2189    \group_begin:
2190    \hook_use:n { scripture / chap / before }
2191    \box_use:N \l__scripture_chap_box
2192    \hook_use:n { scripture / chap / after }
2193    \group_end:
2194    \bool_if:NT \l__scripture_chap_show_verse_bool
2195      {
2196        \__scripture_verse_output:n { 1 }
2197      }
2198    \__scripture_pilcrow_output:
2199    \bool_if:NT \l__scripture_verse_para_bool
2200      {
2201        \hook_gput_next_code:nn { para / after }
2202          {
2203            \int_gset_eq:NN \g__scripture_chap_par_prevgraf_int \prevgraf
2204            \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
2205              { \penalty 10000 }
2206            \hook_gput_next_code:nn { para / before }
2207              {
2208                \int_set_eq:NN \clubpenalty \l__scripture_clubpenalty_saved_int
```

```
2209                        }
2210                    }
2211                }
2212            \hook_gput_next_code:nn { para / begin }
2213                {
2214                    \int_compare:nNnTF \g__scripture_chap_par_prevgraf_int = \c_one_int
2215                        {
2216                            \dim_compare:nNnTF
2217                                { \l_scripture_hanging_leftmargin_tl } < { \g__scripture_chap_width_dim + \l_
2218                                {
2219                                    \parshape 2 ~
2220                                        \dim_eval:n
2221                                            {
2222                                                \@totalleftmargin - \leftmargin + \g__scripture_chap_width_dim +
2223                                                \l__scripture_chap_sep_tl
2224                                            } ~
2225                                        \dim_eval:n
2226                                            {
2227                                                \linewidth + \leftmargin - \g__scripture_chap_width_dim -
2228                                                \l__scripture_chap_sep_tl
2229                                            } ~
2230                                        \@totalleftmargin ~ \linewidth
2231                                }
2232                                { \parshape 1 ~ \@totalleftmargin ~ \linewidth }
2233                            \hook_gput_next_code:nn { para / begin }
2234                                { \parshape 1 ~ \@totalleftmargin ~ \linewidth }
2235                        }
2236                        { \parshape 1 ~ \@totalleftmargin ~ \linewidth }
2237                    \int_gzero:N \g__scripture_chap_par_prevgraf_int
2238                }
2239        }
```

(*End of definition for* __scripture_hanging_chap:n.)

__scripture_hanging_verse:n   Position a \vs in a hanging environment. The verse is set to overlap to the left if at the beginning of a hanging paragraph.

```
2240 \cs_new_protected:Nn \__scripture_hanging_verse:n
2241    {
2242        \mode_if_vertical:TF
2243            {
2244                \tl_set_eq:NN \l_tmpa_tl \l__scripture_verse_sep_tl
2245                \tl_set_eq:NN \l__scripture_verse_sep_tl \l__scripture_hanging_verse_sep_tl
2246                \bool_lazy_and:nnTF
2247                    { \legacy_if_p:n { @newlist } }
2248                    { \g__scripture_suppress_next_pilcrow_bool }
2249                    {
2250                        \strut
2251                        \bool_if:NTF \l__scripture_hanging_verse_right_bool
2252                            {
2253                                \hbox_overlap_left:n
2254                                    {
2255                                        \__scripture_verse_output:n { #1 }
2256                                    }
2257                            }
```

```
2258                    {
2259                      \hbox_overlap_left:n
2260                        {
2261                          \hbox_overlap_right:n
2262                            {
2263                              \__scripture_verse_output:n { #1 }
2264                            }
2265                          \skip_horizontal:N \l__scripture_hanging_leftmargin_tl
2266                        }
2267                    }
2268                }
2269                {
2270                  \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
2271                  \strut
2272                  \bool_if:NTF \l__scripture_hanging_verse_right_bool
2273                    {
2274                      \hbox_overlap_left:n
2275                        {
2276                          \__scripture_verse_output:n { #1 }
2277                        }
2278                    }
2279                    {
2280                      \hbox_overlap_left:n
2281                        {
2282                          \hbox_overlap_right:n
2283                            {
2284                              \__scripture_verse_output:n { #1 }
2285                            }
2286                          \skip_horizontal:N \l__scripture_hanging_leftmargin_tl
2287                        }
2288                    }
2289                  \__scripture_pilcrow_output:
2290                }
2291            \tl_set_eq:NN \l__scripture_verse_sep_tl \l_tmpa_tl
2292          }
2293          { \__scripture_verse_output:n { #1 } }
2294      }
```

(*End of definition for* __scripture_hanging_verse:n.)

__scripture_nohang: Don't hang a paragraph in a hanging environment. Instead the paragraph is indented by \parindent from its normal hanging position.

```
2295 \cs_new_protected_nopar:Nn \__scripture_nohang:
2296   {
2297     \legacy_if:nTF {@newlist}
2298       { \skip_horizontal:n { \l__scripture_hanging_hang_tl } }
2299       { \tex_noindent:D }
2300     \hbox_to_wd:nn { \l__scripture_hanging_parindent_saved_dim } { }
2301     \peek_meaning:NT \vs
2302       {
2303         \bool_gset_true:N \g__scripture_suppress_next_verse_para_bool
2304         \__scripture_set_pilcrow_hook:
2305       }
2306   }
```

(End of definition for `\__scripture_nohang:`*.)*

`\__scripture_hanging_begin:n` Begin a hanging section in a scripture quotation.

```
2307 \cs_new_protected:Nn \__scripture_hanging_begin:n
2308   {
2309     \bool_if:NT \l__scripture_active_inner_bool
2310       {
2311         \msg_error:nn { scripture } { nested-environment }
2312       }
2313     \bool_set_true:N \l__scripture_active_inner_bool
2314     \str_set:Nn \l__scripture_currenvir_str { hanging }
2315     \tl_if_novalue:nF { #1 }
2316       {
2317         \keys_set:nn
2318           { scripture / hanging }
2319           { #1 }
2320       }
2321     \DeclareDocumentCommand { \ch } { s o m }
2322       {
2323         \tl_if_novalue:nF { ##2 }
2324           {
2325             \tl_set:Nn \l__scripture_X_char_tl { ##2 }
2326           }
2327         \__scripture_hanging_ch:n { ##3 }
2328       }
2329     \bool_if:NTF \l__scripture_verse_para_bool
2330       { \cs_set_eq:NN \vs \__scripture_hanging_verse_para:n }
2331       { \cs_set_eq:NN \vs \__scripture_hanging_verse:n }
2332     \cs_set_eq:NN \nohang \__scripture_nohang:
2333     \dim_set_eq:NN \l__scripture_hanging_parindent_saved_dim \parindent
2334     \legacy_if:nTF { @newlist }
2335       { \dim_add:Nn \l__scripture_outer_itemindent_dim \itemindent }
2336       { \dim_zero:N \l__scripture_outer_itemindent_dim }
2337     \cs_set_eq:NN \par \para_end:
2338     \hook_gclear_next_code:n { para/ before }
2339     \list { }
2340       {
2341         \bool_lazy_or:nnF
2342           { \mode_if_vertical_p: }
2343           { \g__scripture_para_mode_vertical_bool }
2344           { \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool }
2345         \dim_zero:N \labelsep
2346         \dim_set:Nn \leftmargin { \l__scripture_hanging_leftmargin_tl + \l__scripture_hangin
2347         \dim_set_eq:NN \rightmargin \l__scripture_hanging_rightmargin_tl
2348         \dim_set:Nn \itemindent { -\l__scripture_hanging_hang_tl - \l__scripture_outer_itemi
2349         \dim_set:Nn \listparindent { -\l__scripture_hanging_hang_tl }
2350         \skip_set_eq:NN \parsep \parskip
2351         \skip_zero:N \partopsep
2352         \dim_compare:nNnTF \parskip > \l__scripture_hanging_aboveskip_tl
2353           { \skip_zero:N \topsep }
2354           { \skip_set:Nn \topsep { \__scripture_skip_diff:NN \l__scripture_hanging_aboveskip
2355         \__scripture_setup_list_noindent:n { -\l__scripture_hanging_hang_tl }
2356       }
2357     \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
```

81

```
2358        { \addvspace { \baselineskip } }
2359      \int_gzero:N \g__scripture_chap_par_prevgraf_int
2360      \item
2361      \relax
2362    }
```

(*End of definition for* `\__scripture_hanging_begin:n`.)

`\__scripture_hanging_end:` End a hanging section in a scripture quotation.

```
2363 \cs_new_protected:Nn \__scripture_hanging_end:
2364   {
2365     \__scripture_calc_final_line_length:
2366     \dim_compare:nNnTF \@outerparskip > \l__scripture_hanging_belowskip_tl
2367       { \skip_zero:N \@topsepadd }
2368       { \skip_set:Nn \@topsepadd { \__scripture_skip_diff:NN \l__scripture_hanging_belowskip
2369     \skip_gset_eq:NN \g__scripture_prev_inner_below_skip \@topsepadd
2370     \legacy_if_set_false:n { @noparlist }
2371     \endlist
2372     \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
2373   }
```

(*End of definition for* `\__scripture_hanging_end:`.)

3.6.4   The `midparachap` environment

`\__scripture_vbox_set_top:Nw` Define missing `\vbox_set_top:Nw` function.

```
2374 \cs_new_protected:Npn \__scripture_vbox_set_top:Nw #1
2375   {
2376     \tex_setbox:D #1 \tex_vtop:D
2377       \c_group_begin_token
2378         \color_group_begin:
2379   }
```

(*End of definition for* `\__scripture_vbox_set_top:Nw`.)

`\l__scripture_mid_para_chap_bool` Flag if setting a paragraph containing a mid-paragraph chapter.

```
2380 \bool_new:N \l__scripture_mid_para_chap_bool
```

(*End of definition for* `\l__scripture_mid_para_chap_bool`.)

`\l__scripture_mid_para_chap_line_box` Store last line before a mid-paragraph chapter.

```
2381 \box_new:N \l__scripture_mid_para_chap_line_box
```

(*End of definition for* `\l__scripture_mid_para_chap_line_box`.)

`\g__scripture_mid_para_chap_prevdepth_dim` Store the `\prevdepth` of the final paragraph in a `midparachap` environment.

```
2382 \dim_new:N \g__scripture_mid_para_chap_prevdepth_dim
```

(*End of definition for* `\g__scripture_mid_para_chap_prevdepth_dim`.)

`scripture_mid_para_chap_totalleftmargin_dim` Save the `\@totalleftmargin` before starting a `midparachap` environment.

```
2383 \dim_new:N \l__scripture_mid_para_chap_totalleftmargin_dim
```

(*End of definition for* `\l__scripture_mid_para_chap_totalleftmargin_dim`.)

`\g__scripture_mid_para_chap_prevgraf_int`   Store the `\prevgraf` of the final paragraph in a `midparachap` environment.

```
2384 \int_new:N \g__scripture_mid_para_chap_prevgraf_int
```

(*End of definition for* `\g__scripture_mid_para_chap_prevgraf_int`.)

`\l_scripture_mid_para_chap_snap_box`   Store unpacked last line before a mid-paragraph chapter.

```
2385 \box_new:N \l__scripture_mid_para_chap_snap_box
```

(*End of definition for* `\l__scripture_mid_para_chap_snap_box`.)

`\l_scripture_mid_para_chap_spacefactor_int`   Store space factor before a mid-paragraph chapter.

```
2386 \int_new:N \l__scripture_mid_para_chap_spacefactor_int
```

(*End of definition for* `\l__scripture_mid_para_chap_spacefactor_int`.)

`\__scripture_mid_para_chap_end_orig:n`
`\__scripture_mid_para_chap_end:n`   Custom `\end` macro for the `midparachap` environment. This is only used when the `verse/para` and/or `compact` options are in use.

```
2387 \cs_set_eq:Nc \__scripture_mid_para_chap_end_orig:n { end ~ }
2388 \cs_new_nopar:Nn \__scripture_mid_para_chap_end:n
2389   {
2390     \__scripture_mid_para_chap_end_orig:n { #1 }
2391     \bool_if:NT \l__scripture_compact_bool
2392       { \c_space_token }
2393     \peek_remove_spaces:n
2394       {
2395         \peek_analysis_map_inline:n
2396           {
2397             \bool_case:n
2398               {
2399                 { \exp_args:No \token_if_letter_p:N { ##1 } }
2400                   { \__scripture_pilcrow_output: }
2401                 { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \vs }
2402                   { \__scripture_set_pilcrow_hook: }
2403               }
2404             \peek_analysis_map_break:n { ##1 }
2405           }
2406       }
2407   }
```

(*End of definition for* `\__scripture_mid_para_chap_end_orig:n` *and* `\__scripture_mid_para_chap_-end:n`.)

`\l_scripture_mid_para_chap_box`
`\l_scripture_mid_para_chap_split_box`
`\l_scripture_mid_para_chap_split_top_box`   Boxes containing a paragraph with a chapter break and its split if needed.

```
2408 \box_new:N \l__scripture_mid_para_chap_box
2409 \box_new:N \l__scripture_mid_para_chap_split_box
2410 \box_new:N \l__scripture_mid_para_chap_split_top_box
```

(*End of definition for* `\l__scripture_mid_para_chap_box`, `\l__scripture_mid_para_chap_split_box`, *and* `\l__scripture_mid_para_chap_split_top_box`.)

`\__scripture_mid_para_drop_chap_output:`   Output a mid-paragraph drop chapter.

```
2411 \cs_new_protected:Nn \__scripture_mid_para_drop_chap_output:
2412   {
2413     \unskip
2414     \int_set_eq:NN \l__scripture_mid_para_chap_spacefactor_int \spacefactor
2415     \strut
2416     \par
2417     \box_set_to_last:N \l__scripture_mid_para_chap_line_box
2418     \hbox_set:Nn \l__scripture_mid_para_chap_snap_box
2419       {
2420         \hbox_unpack_drop:N \l__scripture_mid_para_chap_line_box
2421         \unskip
2422         \unskip
2423         \unpenalty
2424       }
2425     \nointerlineskip
2426     \dim_compare:nNnTF
2427       {
2428         \box_wd:N \l__scripture_mid_para_chap_snap_box +
2429         \g__scripture_chap_width_dim +
2430         \l__scripture_chap_sep_tl
2431       }
2432       <
2433       { \linewidth - \leftskip - \rightskip - 3em }
2434       {
2435         \__scripture_drop_chap_output:
2436         \hbox_unpack_drop:N \l__scripture_mid_para_chap_snap_box
2437         \int_set_eq:NN \spacefactor \l__scripture_mid_para_chap_spacefactor_int
2438         \c_space_tl
2439         \__scripture_verse_output:n { 1 }
2440       }
2441       {
2442         \group_begin:
2443         \noindent
2444         \skip_horizontal:n { -\leftskip }
2445         \hbox_unpack_drop:N \l__scripture_mid_para_chap_snap_box
2446         \skip_zero:N \parfillskip
2447         \par
2448         \group_end:
2449         \__scripture_drop_chap_output:
2450         \bool_if:NT \l__scripture_midparachap_show_verse_bool
2451           {
2452             \__scripture_verse_output:n { 1 }
2453           }
2454       }
2455   }
```

(*End of definition for* `\__scripture_mid_para_drop_chap_output:`.)

`\__scripture_mid_para_chap_begin:n`   Begin a paragraph containing a mid-paragraph chapter.

```
2456 \cs_new_protected:Nn \__scripture_mid_para_chap_begin:n
2457   {
2458     \str_set:Nn \l__scripture_currenvir_str { midparachap }
2459     \tl_if_novalue:nF { #1 }
```

```
2460        {
2461          \keys_set:nn
2462            { scripture / midparachap }
2463            { #1 }
2464        }
2465      \bool_lazy_and:nnT \l__scripture_chap_show_verse_bool \l__scripture_chap_drop_bool
2466        {
2467          \bool_set_true:N \l__scripture_midparachap_show_verse_bool
2468          \bool_set_false:N \l__scripture_chap_show_verse_bool
2469        }
2470      \par
2471      \bool_set_true:N \l__scripture_mid_para_chap_bool
2472      \__scripture_vbox_set_top:Nw \l__scripture_mid_para_chap_box
2473      \dim_zero:N \leftskip
2474      \dim_zero:N \rightskip
2475      \dim_set_eq:NN \l__scripture_mid_para_chap_totalleftmargin_dim \@totalleftmargin
2476      \dim_zero:N \@totalleftmargin
2477      \dim_set_eq:NN \hsize \linewidth
2478      \skip_zero:N \parskip
2479    }
```

(*End of definition for* `\__scripture_mid_para_chap_begin:n`.)

`\__scripture_mid_para_chap_end:` End a paragraph containing a mid-paragraph chapter.

```
2480 \cs_new_protected:Nn \__scripture_mid_para_chap_end:
2481   {
2482     \par
2483     \dim_gset_eq:NN \g__scripture_mid_para_chap_prevdepth_dim \prevdepth
2484     \box_set_to_last:N \l__scripture_mid_para_chap_line_box
2485     \hbox_set:Nn \l__scripture_mid_para_chap_snap_box
2486       {
2487         \hbox_unpack:N \l__scripture_mid_para_chap_line_box
2488         \unskip
2489         \unskip
2490         \unpenalty
2491       }
2492     \dim_gset:Nn \g__scripture_final_line_dim
2493       { \box_wd:N \l__scripture_mid_para_chap_snap_box + \l__scripture_mid_para_chap_totalle
2494     \box_clear:N \l__scripture_mid_para_chap_snap_box
2495     \nointerlineskip
2496     \int_compare:nNnT
2497       \prevgraf < 3
2498       {
2499         \noindent
2500         \skip_horizontal:n { \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl }
2501         \dim_gadd:Nn \g__scripture_final_line_dim { \g__scripture_chap_width_dim + \l__scrip
2502       }
2503     \box_use_drop:N \l__scripture_mid_para_chap_line_box
2504     \vbox_set_end:
```

Attempt to allow the mid-paragraph chapter box to break across pages.

```
2505     \mode_if_inner:TF
2506       {
2507         \noindent
2508         \box_use_drop:N \l__scripture_mid_para_chap_box
```

```
2509            }
2510            {
2511              \dim_compare:nNnTF
2512                { \box_ht_plus_dp:N \l__scripture_mid_para_chap_box }
2513                >
2514                { \pagegoal - \pagetotal - \baselineskip }
2515                {
2516                  \int_set:Nn \vbadness { 10000 }
2517                  \vbox_set_split_to_ht:NNn
2518                    \l__scripture_mid_para_chap_split_box \l__scripture_mid_para_chap_box
2519                    { \pagegoal - \pagetotal - \baselineskip }
2520                  \vbox_set_top:Nn \l__scripture_mid_para_chap_split_top_box
2521                    { \vbox_unpack_drop:N \l__scripture_mid_para_chap_split_box }
2522                  \noindent
2523                  \box_use_drop:N \l__scripture_mid_para_chap_split_top_box
2524                  \goodbreak
2525                  \box_use_drop:N \l__scripture_mid_para_chap_box
2526                }
2527                {
2528                  \noindent
2529                  \box_use_drop:N \l__scripture_mid_para_chap_box
2530                }
2531            }
2532          \par
2533          \dim_set_eq:NN \prevdepth \g__scripture_mid_para_chap_prevdepth_dim
2534          \int_zero:N \prevgraf
2535          \skip_vertical:n { -\parskip - \baselineskip }
2536          \noindent
2537          \skip_horizontal:N \g__scripture_final_line_dim
2538          \int_gset_eq:NN \g__scripture_chap_par_prevgraf_int \g__scripture_mid_para_chap_prevgraf
2539          \hook_gput_next_code:nn { para / before }
2540            {
2541              \legacy_if:nF { @newlist }
2542                {
2543                  \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
2544                    {
2545                      \dim_compare:nNnT \parskip < \baselineskip
2546                        {
2547                          \dim_compare:nNnT { \g__scripture_chap_width_dim + \l__scripture_chap_se
2548                            {
2549                              \hook_gput_next_code:nn { para / begin }
2550                                {
2551                                  \para_omit_indent:
2552                                  \skip_horizontal:n {\g__scripture_chap_width_dim + \l__scripture
2553                                }
2554                            }
2555                        }
2556                    }
2557                  \int_gzero:N \g__scripture_chap_par_prevgraf_int
2558                }
2559              \parshape 1 ~ \@totalleftmargin ~ \linewidth
2560            }
2561      }
```

(*End of definition for* __scripture_mid_para_chap_end:.)

3.6.5 The narrow environment

__scripture_narrow_begin:n   Begin a narrow section in a scripture quotation.

```
2562 \cs_new_protected:Nn \__scripture_narrow_begin:n
2563   {
2564     \bool_if:NT \l__scripture_active_inner_bool
2565       {
2566         \msg_error:nn { scripture } { nested-environment }
2567       }
2568     \bool_set_true:N \l__scripture_active_inner_bool
2569     \str_set:Nn \l__scripture_currenvir_str { narrow }
2570     \tl_if_novalue:nF { #1 }
2571       {
2572         \keys_set:nn
2573           { scripture / narrow }
2574           { #1 }
2575       }
2576     \legacy_if:nTF { @newlist }
2577       { \dim_add:Nn \l__scripture_outer_itemindent_dim \itemindent }
2578       { \dim_zero:N \l__scripture_outer_itemindent_dim }
2579     \cs_set_eq:NN \par \para_end:
2580     \list { }
2581       {
2582         \dim_zero:N \labelsep
2583         \dim_set_eq:NN \leftmargin \l__scripture_narrow_leftmargin_tl
2584         \dim_set_eq:NN \rightmargin \l__scripture_narrow_rightmargin_tl
2585         \dim_set_eq:NN \listparindent \parindent
2586         \bool_lazy_or:nnTF
2587           { \mode_if_vertical_p: }
2588           { \g__scripture_para_mode_vertical_bool }
2589           {
2590             \dim_set:Nn \itemindent { \parindent - \l__scripture_outer_itemindent_dim }
2591             \__scripture_setup_list_noindent:n { \parindent }
2592           }
2593           {
2594             \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
2595             \dim_set:Nn \itemindent { -\l__scripture_outer_itemindent_dim }
2596             \__scripture_setup_list_noindent:n { \c_zero_dim }
2597           }
2598         \skip_set_eq:NN \parsep \parskip
2599         \skip_zero:N \partopsep
2600         \dim_compare:nNnTF \parskip > \l__scripture_narrow_aboveskip_tl
2601           { \skip_zero:N \topsep }
2602           { \skip_set:Nn \topsep { \__scripture_skip_diff:NN \l__scripture_narrow_aboveskip_
2603       }
2604     \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
2605       { \addvspace { \baselineskip } }
2606     \int_gzero:N \g__scripture_chap_par_prevgraf_int
2607     \bool_if:NT \l__scripture_verse_para_bool
2608       { \cs_set_eq:NN \par \__scripture_para_par: }
2609     \item
2610     \relax
2611   }
```

(*End of definition for* __scripture_narrow_begin:n.)

$\__scripture_narrow_end:$ End a narrow section in a scripture quotation.

```
2612 \cs_new_protected:Nn \__scripture_narrow_end:
2613   {
2614     \cs_set_eq:NN \par \para_end:
2615     \__scripture_calc_final_line_length:
2616     \dim_compare:nNnTF \@outerparskip > \l__scripture_narrow_belowskip_tl
2617       { \skip_zero:N \@topsepadd }
2618       { \skip_set:Nn \@topsepadd { \__scripture_skip_diff:NN \l__scripture_narrow_belowskip_
2619     \skip_gset_eq:NN \g__scripture_prev_inner_below_skip \@topsepadd
2620     \legacy_if_set_false:n { @noparlist }
2621     \endlist
2622     \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
2623   }
```

(*End of definition for* __scripture_narrow_end:.)

3.6.6 The poetry environment

scripture/poetry/para/after This hook is an internal hook used to adjust the \parshape within a poetry environment after drop chapter numbers.

```
2624 \hook_new:n { scripture / poetry / para / after }
```

(*End of definition for* scripture/poetry/para/after. *This variable is documented on page* **??**.)

\g__scripture_poetry_end_bool Flag indicating that a poetry environment just ended.

```
2625 \bool_new:N \g__scripture_poetry_end_bool
```

(*End of definition for* \g__scripture_poetry_end_bool.)

\l__scripture_poetry_midparachap_show_verse_bool Flag indicating whether the verse number should be printed immediately after a \ch or not in a poetry environment.

```
2626 \bool_new:N \l__scripture_poetry_midparachap_show_verse_bool
```

(*End of definition for* \l__scripture_poetry_midparachap_show_verse_bool.)

\l__scripture_poetry_chap_indent_dim Store calculated indent and sep when a drop chapter appears in a poetry environment.
\l__scripture_poetry_chap_sep_dim

```
2627 \dim_new:N \l__scripture_poetry_chap_indent_dim
2628 \dim_new:N \l__scripture_poetry_chap_sep_dim
```

(*End of definition for* \l__scripture_poetry_chap_indent_dim *and* \l__scripture_poetry_chap_sep_-dim.)

\l__scripture_poetry_chap_parshape_correction_dim Correction to \parshape in chapter affected lines in a poetry environment.

```
2629 \dim_new:N \l__scripture_poetry_chap_parshape_correction_dim
```

(*End of definition for* \l__scripture_poetry_chap_parshape_correction_dim.)

\l__scripture_poetry_prevgraf_int Store the number of lines a line of poetry takes up.

```
2630 \int_new:N \l__scripture_poetry_prevgraf_int
```

(*End of definition for* \l__scripture_poetry_prevgraf_int.)

__scripture_poetry_text_right:nn Align a word to the right and finish a paragraph, pushing to the next line if there is not space for the word.

```
2631 \cs_new_protected:Nn \__scripture_poetry_text_right:nn
2632   {
2633     {
2634       \unskip
2635       \hfil
2636       \penalty 50
2637       \skip_horizontal:n { #1 }
2638       \hbox:n {}
2639       \nobreak
2640       \hfill
2641       #2
2642       \skip_zero:N \parfillskip
2643       \int_zero:N \finalhyphendemerits
2644     }
2645   }
```

(*End of definition for* __scripture_poetry_text_right:nn.)

__scripture_selah_output: Output Selah at the end of a line.

```
2646 \cs_new_protected:Nn \__scripture_selah_output:
2647   {
2648     \group_begin:
2649     \l__scripture_selah_font_tl
2650     \__scripture_poetry_text_right:nn
2651       { \l__scripture_selah_sep_tl }
2652       { \__scripture_selah_format:n { \l__scripture_selah_text_tl } }
2653     \group_end:
2654   }
```

(*End of definition for* __scripture_selah_output:.)

__scripture_poetry_ch:n Position a chapter in a poetry environment.

```
2655 \cs_new_protected:Nn \__scripture_poetry_ch:n
2656   {
2657     \bool_if:NTF \l__scripture_chap_show_bool
2658       {
2659         \bool_if:NTF \l__scripture_chap_drop_bool
2660           { \__scripture_poetry_chap:n { #1 } }
2661           {
2662             \bool_if:NTF \l__scripture_chap_para_bool
2663               { \__scripture_para_chap:n { #1 } }
2664               { \__scripture_poetry_nodrop_chap:n { #1 } }
2665           }
2666       }
2667       {
2668         \bool_if:NTF \l__scripture_chap_para_bool
2669           { \__scripture_para_chap:n { #1 } }
2670           { \__scripture_poetry_nodrop_chap:n { #1 } }
2671       }
2672   }
```

(*End of definition for* __scripture_poetry_ch:n.)

`\__scripture_poetry_chap:n` Position a drop chapter in a poetry environment.

```
2673 \cs_new_protected:Nn \__scripture_poetry_chap:n
2674   {
2675     \__scripture_drop_chap_set_up:n { #1 }
2676      \hook_gput_next_code:nn { para / after }
2677        {
2678          \hook_gput_next_code:nn { para / after }
2679            { \int_gzero:N \g__scripture_chap_par_prevgraf_int }
2680        }
2681     \cs_if_eq:NNTF \vs \__scripture_poetry_mode_vertical_verse:n
2682       {
2683         \dim_zero:N \l__scripture_poetry_chap_indent_dim
2684         \noindent
2685       }
2686       {
2687         \dim_set_eq:NN \l__scripture_poetry_chap_indent_dim \l__scripture_poetry_indent_tl
2688         \mode_leave_vertical:
2689       }
2690     \dim_set_eq:NN \l__scripture_poetry_chap_sep_dim \l__scripture_chap_sep_tl
2691     \dim_compare:nNnT
2692       { \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl }
2693       <
2694       { \l__scripture_poetry_leftmargin_tl + \l__scripture_poetry_chap_indent_dim }
2695       {
2696         \dim_set:Nn \l__scripture_poetry_chap_sep_dim
2697           {
2698             \l__scripture_poetry_leftmargin_tl + \l__scripture_poetry_chap_indent_dim -
2699             \g__scripture_chap_width_dim
2700           }
2701       }
2702     \hbox_set:Nn \l__scripture_chap_box
2703       {
2704         \box_move_down:nn
2705           { \baselineskip }
2706           {
2707             \hbox_overlap_left:n
2708               {
2709                 \box_use:N \l__scripture_chap_tmp_box
2710                 \skip_horizontal:N \l__scripture_poetry_chap_sep_dim
2711               }
2712           }
2713       }
2714     \box_set_ht:Nn \l__scripture_chap_box { \c_zero_dim }
2715     \box_set_dp:Nn \l__scripture_chap_box { \c_zero_dim }
2716     \dim_set:Nn \l__scripture_poetry_chap_parshape_correction_dim
2717       {
2718         \g__scripture_chap_width_dim + \l__scripture_poetry_chap_sep_dim -
2719         \l__scripture_poetry_leftmargin_tl - \l__scripture_poetry_chap_indent_dim
2720       }
2721     \parshape 2 ~
2722       \dim_eval:n
2723         { \@totalleftmargin + \l__scripture_poetry_chap_parshape_correction_dim } ~
2724       \dim_eval:n
2725         { \linewidth - \l__scripture_poetry_chap_parshape_correction_dim } ~
```

```
2726        \dim_eval:n
2727          { \@totalleftmargin + \l__scripture_poetry_big_indent_tl } ~
2728        \dim_eval:n
2729          { \linewidth - \l__scripture_poetry_big_indent_tl }
```
Set the \parshape for the second line after a drop chapter number.
```
2730        \dim_compare:nNnT
2731          { \g__scripture_chap_width_dim + \l__scripture_poetry_chap_sep_dim }
2732          >
2733          \l__scripture_poetry_leftmargin_tl
2734          {
2735            \hook_gput_next_code:nn { scripture / poetry / para / after }
2736              {
2737                \int_compare:nNnT \l__scripture_poetry_prevgraf_int = 1
2738                  {
2739                    \cs_if_eq:NNTF \vs \__scripture_poetry_mode_vertical_verse:n
```
This \parshape is in effect when the next line is the first half of a poetry verse.
```
2740                      {
2741                        \dim_compare:nNnTF
2742                          { \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl }
2743                          <
2744                          \l__scripture_poetry_leftmargin_tl
2745                          { \dim_zero:N \l__scripture_poetry_chap_parshape_correction_dim }
2746                          {
2747                            \dim_set:Nn \l__scripture_poetry_chap_parshape_correction_dim
2748                              {
2749                                \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl -
2750                                \l__scripture_poetry_leftmargin_tl
2751                              }
2752                          }
2753                        \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
2754                          {
2755                            \parshape 2 ~
2756                              \dim_eval:n
2757                                { \@totalleftmargin + \l__scripture_poetry_chap_parshape_correct
2758                              \dim_eval:n
2759                                { \linewidth - \l__scripture_poetry_chap_parshape_correction_dim
2760                              \dim_eval:n
2761                                { \@totalleftmargin + \l__scripture_poetry_big_indent_tl } ~
2762                              \dim_eval:n
2763                                { \linewidth - \l__scripture_poetry_big_indent_tl }
2764                          }
2765                      }
```
This \parshape is in effect when the next line is the second half of a poetry verse.
```
2766                      {
2767                        \dim_compare:nNnTF
2768                          { \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl }
2769                          <
2770                          { \l__scripture_poetry_leftmargin_tl + \l__scripture_poetry_indent_tl
2771                          { \dim_zero:N \l__scripture_poetry_chap_parshape_correction_dim }
2772                          {
2773                            \dim_set:Nn \l__scripture_poetry_chap_parshape_correction_dim
2774                              {
```

91

```
2775                              \g__scripture_chap_width_dim + \l__scripture_chap_sep_tl -
2776                              \l__scripture_poetry_leftmargin_tl - \l__scripture_poetry_indent
2777                            }
2778                          }
2779                       \parshape 2 ~
2780                         \dim_eval:n
2781                           { \@totalleftmargin + \l__scripture_poetry_chap_parshape_correction_
2782                         \dim_eval:n
2783                           { \linewidth - \l__scripture_poetry_chap_parshape_correction_dim } ~
2784                         \dim_eval:n
2785                           { \@totalleftmargin + \l__scripture_poetry_big_indent_tl } ~
2786                         \dim_eval:n
2787                           { \linewidth - \l__scripture_poetry_big_indent_tl }
2788                     }
2789                 }
2790             }
2791         }
2792     \group_begin:
2793     \hook_use:n { scripture / chap / before }
2794     \box_use:N \l__scripture_chap_box
2795     \hook_use:n { scripture / chap / after }
2796     \group_end:
2797     \bool_lazy_or:nnT \l__scripture_chap_show_verse_bool \l__scripture_poetry_midparachap_sh
2798       { \__scripture_verse_output:n { 1 } }
2799     \hook_use:n { scripture / poetry / pilcrow }
2800   }
```

(*End of definition for* __scripture_poetry_chap:n.)

__scripture_poetry_nodrop_chap:n Format a non-drop chapter for **poetry** environment in a **scripture** quotation.

```
2801 \cs_new_protected:Nn \__scripture_poetry_nodrop_chap:n
2802   {
2803     \cs_gset_nopar:Npn \scripturecurrentchapter { #1 }
2804     \cs_gset_nopar:Npn \scripturecurrentverse { 1 }
2805     \group_begin:
2806     \cs_if_eq:NNT \vs \__scripture_poetry_mode_vertical_verse:n
2807       { \noindent }
2808     \bool_if:NTF \l__scripture_chap_show_bool
2809       {
2810         \hbox_overlap_left:n
2811           {
2812             \hook_use:n { scripture / chap / before }
2813             \group_begin:
2814             \l__scripture_chap_font_tl
2815             \str_if_eq:VnF \l__scripture_chapter_colour_tl { . }
2816               { \exp_args:NV \color_select:n \l__scripture_chapter_colour_tl }
2817             \__scripture_chap_format:n { #1 }
2818             \group_end:
2819             \skip_horizontal:N \l__scripture_chap_sep_tl
2820             \hook_use:n { scripture / chap / after }
2821           }
2822         \bool_lazy_or:nnT \l__scripture_chap_show_verse_bool \l__scripture_poetry_midparacha
2823           { \__scripture_verse_output:n { 1 } }
2824       }
```

```
2825        {
2826            \hook_use:n { scripture / chap / before }
2827            \hook_use:n { scripture / chap / after }
2828            \vs { 1 }
2829        }
2830        \group_end:
2831        \hook_use:n { scripture / poetry / pilcrow }
2832    }
```

(*End of definition for* __scripture_poetry_nodrop_chap:n.)

__scripture_poetry_mode_vertical_verse:n   Format a verse for poetry scripture quotations in vertical mode.

```
2833 \cs_new_protected:Nn \__scripture_poetry_mode_vertical_verse:n
2834    {
2835        \group_begin:
2836        \tl_set_eq:NN \l__scripture_verse_sep_tl \l__scripture_poetry_verse_sep_tl
2837        \hook_gclear_next_code:n { scripture / pilcrow }
2838        \strut
2839        \int_compare:nNnTF \g__scripture_chap_par_prevgraf_int = 1
2840            {
2841                \__scripture_verse_output:n { #1 }
2842            }
2843            {
2844                \bool_if:NTF \l__scripture_poetry_verse_right_bool
2845                    {
2846                        \hbox_overlap_left:n
2847                            {
2848                                \__scripture_verse_output:n { #1 }
2849                            }
2850                    }
2851                    {
2852                        \hbox_overlap_left:n
2853                            {
2854                                \hbox_overlap_right:n
2855                                    {
2856                                        \__scripture_verse_output:n { #1 }
2857                                    }
2858                                \skip_horizontal:N \l__scripture_poetry_leftmargin_tl
2859                            }
2860                    }
2861            }
2862        \group_end:
2863        \bool_if:NF \g__scripture_suppress_next_pilcrow_bool
2864            { \hook_use:n { scripture / poetry / pilcrow } }
2865    }
```

(*End of definition for* __scripture_poetry_mode_vertical_verse:n.)

__scripture_poetry_mode_horizontal_verse:n   Format a verse for poetry scripture quotations in horizontal mode.

```
2866 \cs_new_protected:Nn \__scripture_poetry_mode_horizontal_verse:n
2867    {
2868        \group_begin:
2869        \tl_set_eq:NN \l__scripture_verse_sep_tl \l__scripture_poetry_verse_sep_tl
2870        \int_compare:nNnTF \g__scripture_chap_par_prevgraf_int = 1
2871            {
```

93

```
2872              \__scripture_poetry_mode_vertical_verse:n { #1 }
2873            }
2874            {
2875              \skip_horizontal:n { -\listparindent }
2876              \__scripture_poetry_mode_vertical_verse:n { #1 }
2877              \skip_horizontal:N \listparindent
2878            }
2879          \group_end:
2880        }
```

(*End of definition for* `\__scripture_poetry_mode_horizontal_verse:n`.)

`\__scripture_poetry_par:`   par macro for poetry sections

```
2881  \cs_new_protected:Nn \__scripture_poetry_par:
2882    {
2883      \para_end:
2884      \int_set:Nn \l__scripture_poetry_prevgraf_int { \int_use:N \prevgraf }
2885      \parshape 2 ~
2886        \@totalleftmargin ~
2887        \linewidth ~
2888        \dim_eval:n
2889          { \@totalleftmargin + \l__scripture_poetry_big_indent_tl } ~
2890        \dim_eval:n
2891          { \linewidth - \l__scripture_poetry_big_indent_tl }
2892      \hook_use:n { scripture / poetry / para / after }
2893    }
```

(*End of definition for* `\__scripture_poetry_par:`.)

`\__scripture_obeylines:`   Special version of obeylines which doesn't insert par at end of environment.

```
2894  \group_begin:
2895    \char_set_catcode_active:N \^^M
2896    \cs_gset_protected_nopar:Nn \__scripture_obeylines: %
2897      {%
2898        \char_set_catcode_active:N \^^M%
2899        \cs_set_eq:NN ^^M \__scripture_obeylines_eol:%
2900      }%
2901  \group_end:
2902  \cs_new_protected:Nn \__scripture_obeylines_eol:
2903    {
2904      \mode_if_vertical:TF
2905        {
2906          \cs_set_eq:NN \vs \__scripture_poetry_mode_vertical_verse:n
2907          \hook_gput_next_code:nn { scripture / poetry / para / after }
2908            { \bool_set_false:N \l__scripture_poetry_midparachap_show_verse_bool }
2909        }
2910        { \cs_set_eq:NN \vs \__scripture_poetry_mode_horizontal_verse:n }
2911      \peek_analysis_map_inline:n
2912        {
2913          \bool_case:nF
2914            {
2915              { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \ch }
2916                {
2917                  \bool_set_eq:NN \l__scripture_poetry_midparachap_show_verse_bool \l__scriptu
2918                  \__scripture_poetry_par:
```

94

```
2919                    \peek_analysis_map_break:n { ##1 }
2920                  }
2921              { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \end }
2922                { \peek_analysis_map_break:n { ##1 } }
2923              { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \extraskip }
2924                { \peek_analysis_map_break:n { ##1 } }
2925              { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \nopilcrow }
2926                {
2927                  \__scripture_poetry_par:
2928                  \cs_if_eq:NNT \vs \__scripture_poetry_mode_vertical_verse:n
2929                    { \noindent }
2930                  \peek_analysis_map_break:n
2931                    {
2932                      \__scripture_fix_protrusion:o { ##1 }
2933                    }
2934                }
2935              { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \vs }
2936                {
2937                  \__scripture_poetry_par:
2938                  \cs_if_eq:NNT \vs \__scripture_poetry_mode_vertical_verse:n
2939                    { \noindent }
2940                  \peek_analysis_map_break:n { ##1 }
2941                }
2942              { \exp_args:No \token_if_eq_meaning_p:NN { ##1 } \__scripture_obeylines_eol: }
2943                { \cs_set_eq:NN \vs \__scripture_poetry_mode_vertical_verse:n }
2944            }
2945            {
2946              \__scripture_poetry_par:
2947              \cs_if_eq:NNT \vs \__scripture_poetry_mode_vertical_verse:n
2948                { \noindent }
2949              \bool_if:NT \l__scripture_verse_para_bool
2950                {
2951                  \__scripture_pilcrow_output:
2952                  \bool_set_false:N \l__scripture_verse_para_bool
2953                  \bool_set_false:N \l__scripture_pilcrow_bool
2954                }
2955              \peek_analysis_map_break:n
2956                {
2957                  \__scripture_fix_protrusion:o { ##1 }
2958                }
2959            }
2960        }
2961    }
```

(*End of definition for* `\__scripture_obeylines:`.)

`\__scripture_poetry_begin:` Begin a poetry section in a scripture quotation.

```
2962 \cs_new_protected:Nn \__scripture_poetry_begin:
2963   {
2964     \bool_if:NT \l__scripture_active_inner_bool
2965       {
2966         \msg_error:nn { scripture } { nested-environment }
2967       }
2968     \bool_set_true:N \l__scripture_active_inner_bool
```

```
2969        \str_set:Nn \l__scripture_currenvir_str { poetry }
2970        \bool_gset_false:N \g__scripture_poetry_midparachap_show_verse_bool
2971        \DeclareDocumentCommand { \ch } { s o m }
2972          {
2973            \tl_if_novalue:nF { ##2 }
2974              {
2975                \tl_set:Nn \l_scripture_X_char_tl { ##2 }
2976              }
2977            \__scripture_poetry_ch:n { ##3 }
2978          }
2979        \cs_set_eq:NN \vs \__scripture_poetry_mode_vertical_verse:n
2980        \cs_set_eq:NN \selah \__scripture_selah_output:
2981        \cs_set_protected_nopar:Npn \textright ##1
2982          { \__scripture_poetry_text_right:nn { \l__scripture_text_right_sep_tl } { ##1 } } }
2983        \legacy_if:nTF { @newlist }
2984          { \dim_add:Nn \l__scripture_outer_itemindent_dim \itemindent }
2985          { \dim_zero:N \l__scripture_outer_itemindent_dim }
2986        \cs_set_eq:NN \par \para_end:
2987        \cs_set_eq:NN \noindent \tex_noindent:D
2988        \hook_gput_next_code:nn { scripture / poetry / pilcrow }
2989          {
2990            \__scripture_pilcrow_output:
2991            \bool_set_false:N \l__scripture_pilcrow_bool
2992            \bool_set_false:N \l__scripture_verse_para_bool
2993          }
2994        \list { }
2995          {
2996            \bool_lazy_or:nnF
2997              { \mode_if_vertical_p: }
2998              { \g__scripture_para_mode_vertical_bool }
2999              { \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool }
3000            \dim_zero:N \labelsep
3001            \dim_set_eq:NN \leftmargin \l__scripture_poetry_leftmargin_tl
3002            \dim_set_eq:NN \rightmargin \l__scripture_poetry_rightmargin_tl
3003            \dim_set:Nn \itemindent { \l__scripture_poetry_indent_tl - \l__scripture_outer_itemi
3004            \dim_set_eq:NN \listparindent \l__scripture_poetry_indent_tl
3005            \skip_zero:N \parsep
3006            \skip_zero:N \partopsep
3007            \dim_compare:nNnTF \parskip > { \l__scripture_poetry_aboveskip_tl }
3008              { \skip_zero:N \topsep }
3009              { \skip_set:Nn \topsep { \__scripture_skip_diff:NN \l__scripture_poetry_aboveskip_
3010            \__scripture_setup_list_noindent:n { \l__scripture_poetry_indent_tl }
3011          }
3012        \int_compare:nNnT \g__scripture_chap_par_prevgraf_int = 1
3013          { \addvspace { \baselineskip } }
3014        \int_gzero:N \g__scripture_chap_par_prevgraf_int
3015        \raggedright
3016        \dim_set_eq:NN \parindent \l__scripture_poetry_indent_tl
3017        \item
3018        \relax
3019      }
```

(*End of definition for* __scripture_poetry_begin:.)

`\__scripture_poetry_end:`   End a poetry section in a scripture quotation.

```
3020 \cs_new_protected:Nn \__scripture_poetry_end:
3021   {
3022     \__scripture_calc_final_line_length:
3023     \dim_compare:nNnTF \@outerparskip > { \l__scripture_poetry_belowskip_tl }
3024       { \skip_zero:N \@topsepadd }
3025       { \skip_set:Nn \@topsepadd { \__scripture_skip_diff:NN \l__scripture_poetry_belowskip_
3026     \skip_gset_eq:NN \g__scripture_prev_inner_below_skip \@topsepadd
3027     \legacy_if_set_false:n { @noparlist }
3028     \endlist
3029     \bool_gset_true:N \g__scripture_suppress_next_pilcrow_bool
3030     \hook_gclear_next_code:n { scripture / poetry / pilcrow }
3031     \bool_gset_true:N \g__scripture_poetry_end_bool
3032     \hook_gput_next_code:nn { para / begin }
3033       { \bool_gset_false:N \g__scripture_poetry_end_bool }
3034   }
```

(*End of definition for* `\__scripture_poetry_end:`.)

3.7 Public LaTeX Interface

`\scripturecurrentchapter`
`\scripturecurrentverse`
   Globally stores the most recent argument of `\ch` and `\vs`. These could be used in hooks, for example.

```
3035 \cs_set_nopar:Npn \scripturecurrentchapter { }
3036 \cs_set_nopar:Npn \scripturecurrentverse { }
```

(*End of definition for* `\scripturecurrentchapter` *and* `\scripturecurrentverse`. *These functions are documented on page 7.*)

`\scripturesetup`   #1 : ⟨options⟩ to set.
     Set key-value settings.

```
3037 \NewDocumentCommand { \scripturesetup } { m }
3038   {
3039     \keys_set:nn { scripture } { #1 }
3040   }
```

(*End of definition for* `\scripturesetup`. *This function is documented on page 7.*)

`\scripturestyle`   #1 : ⟨name⟩ of the style.
           #2 : ⟨options⟩ to set for the style.
     Set key-value settings to be applied when using the `style` option with value `name`. You may want to pass `defaults` as the first option to ensure any unset options in your style are expected defaults. The starred version appends the `options` to any existing ones.

```
3041 \NewDocumentCommand { \scripturestyle } { s m m }
3042   {
3043     \__scripture_setup_style:nnn { #1 } { #2 } { #3 }
3044   }
```

(*End of definition for* `\scripturestyle`. *This function is documented on page 7.*)

`\textscripture`   #1 : ⟨reference⟩ of quotation.
           #2 : ⟨options⟩ to set.

#3 : ⟨quotation⟩ text.

Set an inline quotation. This is equivalent to calling the `scripture` environment with the `inline` option set.

```
3045 \NewDocumentCommand { \textscripture } { o o +m }
3046   {
3047     \tl_if_novalue:nTF { #2 }
3048       {
3049         \begin { scripture } [ #1 ] [ inline ]
3050       }
3051       {
3052         \begin { scripture } [ #1 ] [ inline, #2 ]
3053       }
3054     #3
3055     \end { scripture }
3056   }
```

(*End of definition for* `\textscripture`. *This function is documented on page* *8.*)

scripture #1 : ⟨reference⟩ of quotation.

#2 : ⟨options⟩ to set.

Format a scripture quotation with optional reference.

```
3057 \NewDocumentEnvironment { scripture } { o o }
3058   {
3059     \__scripture_begin:nn { #1 } { #2 }
```

#1 : ⟨options⟩ to set.

center

```
3060     \DeclareDocumentEnvironment { center } { o }
3061       {
3062         \__scripture_center_begin:n { ##1 }
3063         \__scripture_reference_start_peek:
3064       }
3065       { \__scripture_center_end: }
```

#1 : ⟨options⟩ to set.

flushright

```
3066     \DeclareDocumentEnvironment { flushright } { o }
3067       {
3068         \__scripture_flushright_begin:n { ##1 }
3069         \__scripture_reference_start_peek:
3070       }
3071       { \__scripture_flushright_end: }
```

#1 : ⟨options⟩ to set.

hanging

```
3072     \DeclareDocumentEnvironment { hanging } { o }
3073       {
3074         \__scripture_hanging_begin:n { ##1 }
3075         \__scripture_reference_start_peek:
3076       }
3077       { \__scripture_hanging_end: }
```

midparachap

```
3078     \DeclareDocumentEnvironment { midparachap } { o }
3079       {
3080         \__scripture_mid_para_chap_begin:n { ##1 }
3081         \__scripture_reference_start_peek:
3082       }
3083       { \__scripture_mid_para_chap_end: }
```

#1 : ⟨options⟩ to set.

```
3084       \DeclareDocumentEnvironment { narrow } { o }
3085         {
3086           \__scripture_narrow_begin:n { ##1 }
3087           \__scripture_reference_start_peek:
3088         }
3089         { \__scripture_narrow_end: }
```

#1 : ⟨options⟩ to set.

The `poetryaux` environment is required because `\__scripture_obeylines:` breaks the reading of optional arguments.

```
3090       \DeclareDocumentEnvironment { poetryaux } { o }
3091         {
3092           \tl_if_novalue:nF { ##1 }
3093             {
3094               \keys_set:nn
3095                 { scripture / poetry }
3096                 { ##1 }
3097             }
3098           \__scripture_poetry_begin:
3099           \__scripture_reference_start_peek:
3100         }
3101         { \__scripture_poetry_end: }
3102       \DeclareDocumentEnvironment { poetry } { }
3103         {
3104           \__scripture_obeylines:
3105           \poetryaux
3106         }
3107         { \endpoetryaux }

3108       \bool_if:NT \l__scripture_inline_bool
3109         {
3110           \tl_set:Nn \l__scripture_aboveskip_tl { 0pt }
3111           \tl_set:Nn \l__scripture_belowskip_tl { 0pt }
3112           \tl_set:Nn \l__scripture_leftmargin_tl { 0pt }
3113           \tl_set:Nn \l__scripture_rightmargin_tl { 0pt }
3114           \cs_set_eq:NN \__scripture_ref_format:n \__scripture_inline_ref_format:n
3115           \tl_set_eq:NN \l__scripture_ref_sep_tl \l__scripture_inline_ref_sep_tl
3116           \cs_set_eq:NN \__scripture_version_format:n \__scripture_inline_version_format:n
3117           \tl_set_eq:NN \l__scripture_version_delim_tl \l__scripture_inline_version_delim_tl
3118           \bool_set_false:N \l__scripture_chap_para_bool
3119           \bool_set_true:N \l__scripture_compact_bool
3120           \bool_set_false:N \l__scripture_heading_show_bool
3121         }
3122       \bool_if:NT \l__scripture_chap_para_bool
3123         {
3124           \RenewDocumentEnvironment { midparachap } { o } { \ignorespaces } { \unskip }
3125         }
3126       \bool_if:NT \l__scripture_verse_para_bool
3127         {
3128           \cs_set_eq:NN \vs \__scripture_verse_para_output:n
3129           \cs_set_eq:NN \par \__scripture_para_par:
3130           \cs_set_eq:NN \@par \__scripture_para_par:
3131           \cs_set_eq:NN \@doendpe \__scripture_doendpe:
```

```
3132          \cs_set_eq:NN \noindent \__scripture_para_noindent:
3133          \cs_set_protected:Npn \\
3134            {
3135              \mode_if_horizontal:TF
3136                { \unskip }
3137                { \@nolnerr }
3138            }
3139          \RenewDocumentEnvironment { midparachap } { o }
3140            {
3141              \str_set:Nn \l__scripture_currenvir_str { midparachap }
3142              \cs_set_eq:NN \noindent \tex_noindent:D
3143              \cs_set_eq:cN { end ~ } \__scripture_mid_para_chap_end:n
3144              \__scripture_reference_start_peek:
3145            }
3146            {
3147              \para_end:
3148              \bool_gset_false:N \g__scripture_suppress_next_pilcrow_bool
3149            }
3150        }
3151      \bool_if:NT \l__scripture_compact_bool
3152        {
3153          \cs_set_eq:NN \noindent \__scripture_para_noindent:
3154          \bool_if:NF \l__scripture_verse_para_bool
3155            {
3156              \bool_set_false:N \l__scripture_chap_drop_bool
3157              \bool_set_false:N \l__scripture_chap_drop_llap_bool
3158              \bool_set_false:N \l__scripture_chap_drop_smash_bool
3159              \bool_set_false:N \l__scripture_chap_para_bool
3160              \cs_set_eq:NN \vs \__scripture_verse_compact_output:n
3161            }
3162          \bool_gset:Nn \g__scripture_mode_vertical_inner_bool
3163            { \mode_if_vertical_p: }
3164          \RenewDocumentEnvironment { center } { o }
3165            {
3166              \str_set:Nn \l__scripture_currenvir_str { center }
3167              \__scripture_reference_start_peek:
3168            }
3169            { \__scripture_compact_end_inner: }
3170          \RenewDocumentEnvironment { flushright } { o }
3171            {
3172              \str_set:Nn \l__scripture_currenvir_str { flushright }
3173              \__scripture_reference_start_peek:
3174            }
3175            { \__scripture_compact_end_inner: }
3176          \RenewDocumentEnvironment { hanging } { o }
3177            {
3178              \str_set:Nn \l__scripture_currenvir_str { hanging }
3179              \__scripture_reference_start_peek:
3180            }
3181            { \__scripture_compact_end_inner: }
3182          \RenewDocumentEnvironment { midparachap } { o }
3183            {
3184              \str_set:Nn \l__scripture_currenvir_str { midparachap }
3185              \cs_set_eq:cN { end ~ } \__scripture_mid_para_chap_end:n
```

```
3186            \__scripture_reference_start_peek:
3187          }
3188        { \__scripture_compact_end_inner: }
3189      \RenewDocumentEnvironment { narrow } { o }
3190        {
3191          \str_set:Nn \l__scripture_currenvir_str { narrow }
3192          \__scripture_reference_start_peek:
3193        }
3194        { \__scripture_compact_end_inner: }
3195      \RenewDocumentEnvironment { poetry } { o }
3196        {
3197          \str_set:Nn \l__scripture_currenvir_str { poetry }
3198          \__scripture_reference_start_peek:
3199        }
3200        { \__scripture_compact_end_inner: }
3201      \cs_set_eq:NN \\ \unskip
3202      \cs_set_eq:NN \par \__scripture_compact_par:
3203      \cs_set_eq:NN \extraskip \relax
3204      \cs_set_eq:NN \nohang \relax
3205      \cs_set:Npn \selah
3206        {
3207          \group_begin:
3208          \l__scripture_selah_font_tl
3209          \c_space_tl
3210          \__scripture_selah_format:n { \l__scripture_selah_text_tl }
3211          \c_space_tl
3212          \group_end:
3213        }
3214      \cs_set_protected_nopar:Npn \textright ##1
3215        {
3216          \unskip
3217          \c_space_tl
3218          ##1
3219        }
3220      }
3221    \__scripture_reference_start_peek:
3222    }
3223    {
3224      \bool_lazy_or:nnT \l__scripture_compact_bool \l__scripture_verse_para_bool
3225        { \cs_set_eq:NN \par \para_end: }
3226      \__scripture_end:
3227    }
```

(*End of definition for* scripture *and others. These functions are documented on page 11.*)

```
3228  %</package>
```

Change History

101

102

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

109

111

112

113

114

116