



# ἔκδοσις

Typesetting TEI xml-Compliant Critical Editions

Robert Alessi

<mailto:alessi@robertalessi.net>

2022/05/15 – v1.5-dev

## Contents

<b>License and disclaimer</b>	<b>2</b>	6.2 Single-Layer Apparatus Criticus . . . . .	<b>36</b>
<b>1 Introduction</b>	<b>3</b>	6.3 Multiple-Layer Apparatus Criticus . . . . .	<b>38</b>
1.1 Requirements . . . . .	4	6.3.1 Declaring Additional Layers . . . . .	38
1.2 Features . . . . .	4	6.3.2 Laying Out Layers With The Optional Argument of <code>\DeclareApparatus</code> . . . . .	38
<b>2 The Basics of ekdosis</b>	<b>6</b>	<b>7 Inserting Notes in Multiple-Layer Apparatus</b>	<b>39</b>
2.1 Loading the Package—General Options . . . . .	6	7.1 Variant Readings . . . . .	39
2.2 Setup . . . . .	7	7.2 Other Notes for Comments, Sources or Testimonia . . . . .	40
2.3 Using a Configuration File . . . . .	8	<b>8 Poetry</b>	<b>43</b>
2.4 Witnesses, Hands, Sources, Scholars & Shorthands . . . . .	8	8.1 The Standard verse Environment . . . . .	43
2.4.1 Printing Formatted Witnesses — <i>Conspetus Siglorum</i> . . . . .	12	8.2 The verse Package . . . . .	44
2.5 Editing a Single Text . . . . .	13	<b>9 Lineation Settings</b>	<b>51</b>
2.6 Indicating Subvariation in Apparatus Entries . . . . .	18	<b>10 Languages</b>	<b>53</b>
2.6.1 Implicit Grouping . . . . .	19	10.1 Languages Written From Right to Left . . . . .	54
2.6.2 Explicit Grouping . . . . .	20	10.2 Using <code>arabluatex</code> . . . . .	56
<b>3 Emendations and Conjectures</b>	<b>20</b>	<b>11 Divisions of the Body</b>	<b>57</b>
3.1 Editorial Addition and Deletion . . . . .	22	11.1 L <sup>A</sup> T <sub>E</sub> X Standard Divisions . . . . .	58
<b>4 Lacunae</b>	<b>26</b>	11.2 TEI Un-numbered Divisions . . . . .	59
<b>5 Alignment of Parallel Texts</b>	<b>27</b>	11.3 Headers and Footers . . . . .	61
5.1 Alignment of Several Texts . . . . .	30	<b>12 The Tricks of the Trade</b>	<b>64</b>
5.1.1 Appending Hooks to Environments . . . . .	32	12.1 The Oscillating Problem . . . . .	65
5.2 Laying Out Parallel Texts . . . . .	32	12.2 <i>Variae Quaestiones</i> . . . . .	66
5.2.1 Columns and Gutters . . . . .	33	<b>13 TEI xml Output</b>	<b>67</b>
5.2.2 Marginal Notes . . . . .	33		
5.2.3 Regular Footnotes . . . . .	34		
<b>6 Laying Out the Apparatus Criticus</b>	<b>34</b>		
6.1 General Hooks . . . . .	34		

13.1 Requesting TEI <code>xml</code> Output . . . . .	68	<b>16 References</b>	<b>89</b>
13.2 General Principles . . . . .	68		
13.3 Routine L <sup>A</sup> T <sub>E</sub> X Commands and Environments . . . . .	70	<b>17 Sample: C. J. Caesar, <i>Gallic War</i>, VI, XIII.1</b>	<b>90</b>
13.4 Processing New Commands or Environments . . . . .	72	17.1 <code>.tex</code> Source File . . . . .	90
13.5 Inserting Code in the TEI <code>xml</code> Output File . . . . .	74	17.2 TEI <code>xml</code> Output . . . . .	93
13.6 Specific TEI Modules . . . . .	75	<b>18 Arabic Sample File</b>	<b>97</b>
13.7 References to Cited Works . . . . .	76	<b>19 Implementation</b>	<b>98</b>
13.8 Citation Commands . . . . .	79		
<b>14 Future Work</b>	<b>81</b>	<b>20 Change History</b>	<b>138</b>
<b>15 GNU Free Documentation License</b>	<b>82</b>	<b>21 Index</b>	<b>139</b>

## List of listings

1 The “Peter/John” basic example . . . . .	5	7 Caesar’s <i>Gallic War</i> , VI, 13.1 . . . . .	28
2 Conspectus Siglorum of Caesar’s <i>Gallic War</i> . . . . .	12	8 Poetry: Shakespeare’s Sonnet 1 . . . . .	48
3 The “Peter/John” full example . . . . .	17	9 Multilingual editions with <code>babel</code> only . . . . .	54
4 The “Peter/John” full example: TEI <code>xml</code> output . . . . .	18	10 Multilingual editions with <code>babel</code> and <code>polyglossia</code> . . . . .	55
5 Emendations, conjectures and corrections . . . . .	24	11 <code>ekdosis</code> and <code>arabluatex</code> . . . . .	56
6 Cicero, <i>ad Atticum epistulae</i> , VII, 9.4 . . . . .	26	12 Divisions of the body text . . . . .	60
		13 “Mirrored” paired facing pages . . . . .	64

### Abstract

`ekdosis` is a LuaL<sup>A</sup>T<sub>E</sub>X package designed for multilingual critical editions. It can be used to typeset texts and different layers of critical notes in any direction accepted by LuaT<sub>E</sub>X. Texts can be arranged in running paragraphs or on facing pages, in any number of columns which in turn can be synchronized or not. In addition to printed texts, `ekdosis` can convert `.tex` source files so as to produce TEI `xml`-compliant critical editions. Database-driven encoding under L<sup>A</sup>T<sub>E</sub>X then allows extraction of texts entered segment by segment according to various criteria: main edited text, variant readings, translations or annotated borrowings between texts. It is published under the terms of the GNU General Public License (GPL) version 3.

## License and Disclaimer

### License Applicable to this Document

fdl1.3 This document is part of the work: The `ekdosis` Package.

© 2020–2022 Robert Alessi.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by

the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”. To view a copy of this license, visit: <https://www.gnu.org/licenses/fdl-1.3.html>. (See below [sect. 15](#) on page 82.)

## License Applicable to the Software

ekdosis — Typesetting TEI  $\text{xm1}$ -Compliant Critical Editions © 2020–2022 Robert Alessi.

Please send error reports and suggestions for improvements to Robert Alessi:

- email: <mailto:RobertAlessi@alessi@robertalessi.net>
- website: <http://www.ekdosis.org>
- development: <http://git.robertalessi.net/ekdosis>
- comments, feature requests, bug reports: <http://www.ekdosis.org/issues>

gpl3+ This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

This release of `ekdosis` consists of the following source files:

- `ekdosis.ins`
- `ekdosis.dtx`
- `ekdosis.el`
- `Makefile`


`\eKd` The distinctive emblem of `ekdosis` is made of the three Greek letters  $\epsilon$ ,  $\chi$  and  $\delta$ , like *New feature v1.5* so:  $\epsilon\chi\delta$ . Provided that the font used includes these Unicode glyphs, it is produced by the command `\eKd` and best printed with the *Old Standard* Greek font.<sup>1</sup>

## 1 Introduction

THE READER will find here, by way of introduction, a summarized version of the first part of an article published in the *Journal of Data Mining and Digital Humanities* as a contribution to a Digital Humanities workshop held at Stanford University (April 15, 2019).<sup>2</sup>

The name of this package, `ekdosis`, derives from a Greek action noun— $\epsilon\chi\delta\omicron\sigma\iota\varsigma$ —the meaning of which is: “publishing a book”, and also in concrete sense: “a publication, treatise”. For us moderns, this term refers to a long tradition of scholarly work consisting in establishing from manuscript evidence the texts of Greek and Latin classics that were handled down through the Middle Ages to the time of the first printed editions. Of course, this definition is extendible to other languages as well. The basic premise is that critical editions exhibit reconstructed texts from manuscript evidence either under the title of the edited text (direct tradition) or from explicit citations or parallel passages or translations in other languages (indirect tradition).

1. Robert Alessi, *Old Standard: A Unicode Font for Classical and Medieval Studies* (version 2.6) [Based on Alexey Kryukov’s original Old Standard] (Dec. 18, 2020), <http://ctan.org/pkg/oldstandard>.

2. Robert Alessi, “`ekdosis`: Using  $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$  for Producing TEI  $\text{xm1}$ -Compliant Critical Editions and Highlighting Parallel Writings,” *Journal of Data Mining and Digital Humanities: Collecting, Preserving, and Disseminating Endangered Cultural Heritage for New Understandings through Multilingual Approaches* (Nov. 2020), [jdmhdh: 6536](#). 

Whether in print or digital, critical editions come with an apparatus criticus in which is mentioned all the evidence that was used to build the edited text. Arguably, it is precisely on this common point that the two kind of editions part ways for reading a traditional, well written apparatus criticus is only meant for experienced readers. Getting oneself familiarized with its many conventional rules is not unrelated to learning a language, equipped with technical terms, grammar rules and style embellishments, which came into existence out of over three centuries of scholarly attainments. Nevertheless, whereas this language is immediately accessible to human mind’s ability to use language and interpret conventional symbols, it is quite inaccessible to a computer unless every item of information has been encoded in the rather dumb format that is suited to machines.

On the other hand, editions in print have their own limitations. For example, every detail that editors of classical texts decide to discard to save space, regardless to its relevance to the purpose of the edition, is lost permanently as in the case of dialectal coloring of ancient books. Furthermore, passages collected as indirect tradition are only available as references in the *apparatus testium* and cannot be referred to the original text. As a result, the reader is refrained from bestowing attention upon major parallel passages to understand better difficult passages.

To conclude on these issues, print publications and digital editions are often contrasted as they belonged to two different worlds.<sup>3</sup> It is commonly said that the content of editions in print is the result of the binding of the book itself as an object, whereas digital editions, in which format and presentation are by definition separated from content, are free from limitations coming from such bindings. To sum up from the foregoing considerations, this statement is likely to be qualified: as already seen above, the apparatus criticus must be looked at as a brilliant production of mind refined by centuries of scholarly tradition—and surely tradition must go on—arguably not as compact paragraphs that require special and painful training to be ‘decoded’. On the other hand, what editions in print do not provide are what Donald J. Mastronarde and Richard J. Tarrant have called “actionable texts for use in digital research”,<sup>4</sup> namely database-driven texts allowing the reader to select annotations and display or arrange translations, parallel passages or borrowings in a variety of ways. ekdosis can be seen as an attempt at combining the two approaches.

## 1.1 Requirements

Please refer to [sect. 19 on page 99](#).

## 1.2 Features

A list of the main features of ekdosis follows:—

- (a) *Multilingual critical editions*: ekdosis can be used to typeset any number of texts in any direction accepted by LuaTeX. Running paragraphs of text can be arranged in any number of columns, either on single or facing pages, which in turn can be synchronized or not. ekdosis is also suitable for complex layouts as in the case of Arabic poetry or images where three-way alignment is required, or diagrams, *&c.*
- (b) *Apparatus criticus*: Edited texts can receive multiple layers of apparatus, e.g. apparatus criticus (to record variant readings), apparatus fontium (to collect references to texts quoted or cited in the edited text), apparatus testium (to collect testimonia or

---

3. For a good illustration of this point, see Digital Latin Library, “Textual Criticism,” <https://digitallatin.org/library-digital-latin-texts/textual-criticism>, accessed May 24, 2020, “Content, not Display.”

4. Donald J. Mastronarde and Richard J. Tarrant, “Review: Guidelines for Encoding Critical Editions for the Library of Digital Latin Texts,” Society for Classical Studies (Dec. 4, 2017), <https://classicalstudies.org/scs-blog/donald-j-mastronarde/review-guidelines-encoding-critical-editions-library-digital-latin>.

parallel passages), or any kind of short notes to be printed on the same page as the edited text, *etc.*

- (c) TEI xml output: `ekdosis` can be instructed to output both PDF and TEI xml files at the same time.
- (d) *Database-driven encoding* under L<sup>A</sup>T<sub>E</sub>X of texts entered segment by segment allows for alignment of parallel texts from multilingual corpora.

Before going into detail, the following simple example will give the reader a general idea of the method of encoding with `ekdosis` authoritative texts composed of lemmata, in a way that is very close to TEI xml encoding:—

Listing 1: The “Peter/John” basic example

```
1 \begin{ekdosis}
2   I
3   \app{
4     \lem{saw}
5     \rdg{met}
6   }
7   my friend \app{\lem{Peter}\rdg{John}} at the station yesterday.
8 \end{ekdosis}
```

PDF output:—

1 I saw my friend Peter at the station yesterday.

---

I saw] met Peter] John

TEI xml output:—

```
<p>I
<app>
  <lem>saw</lem>
  <rdg>met</rdg>
</app>my friend
<app>
  <lem>Peter</lem>
  <rdg>John</rdg>
</app>at the station yesterday.</p>
```

As can be seen from [listing 1](#), the edition text is inserted in the `ekdosis` environment (ll. 1 to 8). Then two `\app{apparatus entry}` commands (ll. 3 and 7) contain the lemma (`\lem{lemma}`), namely the reading that is accepted by the editor, and at least one variant reading (`\rdg{reading}`), ll. 5 and 7). As the listing shows, the editor is free to lay out the code in a legible manner to the eye: the first lemma above spans several lines whereas the second one is written in sequence without spaces.

In the PDF output, the edition text is printed in the upper part of the page, above the line, and naturally shows the accepted readings. The margins are used for numeration. In the apparatus criticus, below the line, reference to the text is made by specifying the number of the line and if several entries refer to the same line, numbers are not repeated. Instead, entries are separated from one another by a broad horizontal space. Finally, a square bracket is used inside entries to distinguish the lemma from the variant readings.

Furthermore, as said above, if a TEI xml output be required, `ekdosis` compiles an additional .xml file an excerpt of which is provided above.

## 2 The Basics of ekdosis

### 2.1 Loading the Package—General Options

ekdosis is loaded in the preamble like so:—

```
\usepackage{ekdosis}
```

ekdosis may be loaded with five optional ‘named arguments’ either of which is set using the syntax  $\langle key \rangle = \langle value \rangle$ . The description of the optional arguments follows.



The reader is invited to refer to the relevant sections of this documentation for more information on how to use these options.

layout `layout=float|footins|keyfloat|fitapp` Default: float

(a) `layout=float` By default, layers of critical notes are inserted as floating environments to be printed at the bottom of the pages.

(b) `layout=footins` This can be set to insert critical notes in the default footnote block which can be considered to be a special kind of float that is printed at the bottom of the pages. In this case, the apparatus criticus will be inserted between regular numbered footnotes, but will carry no footnote mark of its own.

*New feature v1.3* (c) `layout=keyfloat` does approximately the same as the default option `layout=float` but uses the `keyfloat` package<sup>5</sup> to generate the floating environments to be used as containers for critical footnotes. This way, the keys and values provided by this package<sup>6</sup> may be used to achieve such effects as append additional, informative text below the apparatus, draw a line around the apparatus block or change its width.<sup>7</sup>

*New feature v1.3* (d) `layout=fitapp` As described below in [sect. 12.1 on page 65](#), “[The Oscillating Problem](#)”, ekdosis may oscillate indefinitely between different sets of page decisions when one or more apparatus entries attached to the last lines of the edition text on a given page do not fit in the apparatus block. The reader will find in this section of the documentation a detailed account of several ways to circumvent this issue. Alternatively, or rather conjointly with those ways, `layout=fitapp` can be used to instruct ekdosis to scale down the characters of the apparatus block so that the contentious entries can fit. This mechanism uses the “fitting” library provided by the `tcolorbox` package.<sup>8</sup> When this option is set, the apparatus criticus grows normally until a predefined height is reached. This height is set to  $0.5\text{\texttt{textheight}}$  by default.<sup>9</sup> From this point on, the apparatus block ceases to grow; rather, the size of the characters is reduced to allow for additional entries. As a consequence of this rationale, the total number of entries on a given page must not be too high. It is therefore advisable to use `layout=fitapp` conjointly with `maxentries` as described below on pages [35](#) and [39](#) and in [sect. 12.1 on page 65](#).



If used appropriately, this mechanism gives excellent typographical results, notably with complex edition texts of which the entries in the associated apparatus can be quite abundant in number. It may even put an end to the “[oscillating problem](#)” in most of the cases. However, as suitable it may be for high quality typeset texts and final, camera-ready copies, its benefit comes at the expense of slowing down the

5. Brian Dunn, *The Keyfloat package* (version 2.06) [Provides a key/value interface for generating floats] (June 29, 2021), <https://ctan.org/pkg/keyfloat>.

6. See *ibid.*, sect. 2.3, pp. 13–16.

7. See below, on page [35](#) for more information.

8. Thomas F. Sturm, *The Tcolorbox package* (version 4.51) [Coloured boxes, for LaTeX examples and theorems, etc] (June 14, 2021), <https://ctan.org/pkg/tcolorbox>, sect. 22, pp. 438–49.

9. Of course, this height can be modified. See below on page [35](#) for details.

compilation process. Yet looser algorithms can be selected when speed must prevail over quality for intermediate or draft copies.<sup>10</sup>

`divs`      `divs=ekdosis|latex` Default: ekdosis

In many occasions, L<sup>A</sup>T<sub>E</sub>X standard textual divisions do not meet the specific requirements of classical and literary texts, the divisions of which may depend on many different received traditions. `ekdosis` provides a flexible mechanism in which format and presentation have been carefully separated from content. It is designed to build un-numbered TEI divisions allowed to nest recursively.<sup>11</sup> However, if `divs` be set to `latex`, L<sup>A</sup>T<sub>E</sub>X standard textual divisions can be used and will be translated into TEI numbered `<div>` elements.



It must be noted that the two styles are mutually exclusive.

`poetry`      `poetry=verse` Default: not set  
*New feature v1.2* `poetry=verse` instructs `ekdosis` to load and use the facilities supplied by the `verse` package for the typesetting of lines of poetry.<sup>12</sup> The `ekdverse` environment must then be used instead of the `verse` environment that is provided by the `verse` package as described below in [sect. 8.2 on page 44](#).

`parnotes`      `parnotes=true|false|roman` Default: not set  
This named argument does not need a value as it defaults to `true` if used. Apparatus criticus typeset by `ekdosis` may contain notes and footnotes. The latter can be laid out as paragraphed notes below the block of critical notes by means of the `parnotes` package. Additionally, `parnotes=roman` prints these footnotes numbered with Roman numerals.

`teiexport`      `teiexport=true|false|tidy` Default: not set  
This named argument does not need a value as it defaults to `true` if used. If `teiexport` be set to `true`, `ekdosis` is instructed to output both PDF and TEI `xml` files at the same time. By default, the TEI file will receive the same basename as the `.tex` source file, suffixed with `-tei.xml`. The raw `.xml` file that is produced by `ekdosis` can be further processed by the `tidy` console application.<sup>13</sup> To make this happen, `tidy` must be installed and the `.tex` source file must be compiled with the `--shell-escape` facility so that spawning programs from L<sup>A</sup>T<sub>E</sub>X can be allowed.<sup>14</sup>

As an example, the following line loads `ekdosis` and instructs it to output a TEI `xml` file (in addition to the PDF one) and to use `parnotes` to format with Roman numerals the footnotes that are inserted in the apparatus criticus:—

```
\usepackage[teiexport, parnotes=roman]{ekdosis}
```

## 2.2 Setup

`\ekdsetup` Starting from v1.3, `\ekdsetup` can be used to specify options that affect the general behavior of `ekdosis`. `\ekdsetup` is a preamble-only command. It accepts the following `key=value` options the number of which is expected to increase as `ekdosis` grows:

`showpagebreaks` `showpagebreaks=true|false` Initially: false; Default: true

This named argument, which defaults to `true` if used without value, has specific marks printed in the margins so as to spot with a rapid cast of the eye the locations of conditional page breaks generated by the `\ekdpb` command described below on [page 65](#). By default, page breaks generated by `\ekdpb` are identified by the string `spb`—for “soft” page

<sup>10</sup>. See below on [page 35](#) for more information.

<sup>11</sup>. See below, [sect. 11 on page 57](#).

<sup>12</sup>. `verse` does not need to be set if the memoir class be used. See [sect. 8.2 on page 44](#) for more detail.

<sup>13</sup>. See <http://www.html-tidy.org>.

<sup>14</sup>. See <https://texfaq.org/FAQ-spawnprog> for more information on how to do this.



break—whereas those generated by `\ekdpb*` are identified by `hpb`—for “hard” page break. Furthermore, when `\ekdpb` triggers no page break, the marker is printed between square brackets, like so: `[spb]`. In this way, inoperative `\ekdpb` can be easily spotted and removed.

`spbmk` `spbmk=<string>` Default: spb

`spbmk` is used to change the string associated to “soft” page breaks.

`hpbmk` `hpbmk=<string>` Default: hpb

`hpbmk` is used to change the string associated to “hard” page breaks.

As an example, what follows has “soft” page breaks printed in blue and “hard” page breaks printed in red:—

```
\ekdsetup{
  showpagebreaks,
  spbmk = \textcolor{blue}{spb},
  hpbmk = \textcolor{red}{hpb}
}
```

## 2.3 Using a Configuration File

Complex editions may use a large number of witnesses, sources and scholars. It may also be required to define a multiple-layer apparatus criticus, several text environments to be aligned and quite a number of new commands. `ekdosis` provides a convenient way to avoid overloading the document preamble: all the settings related to the critical edition can be gathered in a separate configuration file named `\jobname-ekd.cfg`. If such a file can be found, its contents are automatically read and used by `ekdosis`.

## 2.4 Witnesses, Hands, Sources, Scholars & Shorthands

**Terminology** Strictly speaking, the term “witness” should apply to any manuscript evidence dating back to the Middle Ages used by the editor to establish the edition text. That said, editors often consult many other types of documents, such as modern editions, articles, notes, correspondence and the like, all of which fall into the category of “sources”. Furthermore, unpublished conjectures are also taken into account, not to mention the corrections and emendations that are proposed in many places by the editor of the text. As it is necessary to refer to scholars as individuals, “scholars” naturally emerges as a third category.

Any reference that is to be used in the apparatus criticus must be “declared” in the preamble beforehand, namely: manuscript sigla (either for single manuscripts or manuscript families, primary or later hands, *&c.*), abbreviated last names of sources and scholars. To that effect, `ekdosis` provides the following preamble-only commands:—

`\DeclareWitness` **Witnesses** `\DeclareWitness{\<unique id>}{\<rendition>}{\<description>}[\<options>]`

This command requires three mandatory arguments enclosed between curly braces used to specify consecutively:

- (a) The unique identifier of the witness to be used both in the `.tex` source file and as `xml:id` in the TEI `xml` output if any.<sup>15</sup>
- (b) The rendition to be used in the printed apparatus criticus, which also will be found within the `<sourceDesc>` element of the TEI header where the description of the witness occurs, within a `<abbr type="siglum">` element.

---

<sup>15</sup> See on page 70 for more information.



(c) A basic description of the manuscript to be found in a typical printed *Conspectus Siglorum*, namely: the name of the manuscript followed by its call number.

Finally, the optional argument of `\DeclareWitness` accepts a comma-separated list of the following “name=value” arguments the first six of which are used to collect items of information to be found within the `<msIdentifier>` element in the TEI header:<sup>16</sup>—

`settlement settlement=<name>`: The name of a city or administrative unit.  
`institution institution=<name>`: The name of an institution such as a university or library.  
`repository repository=<name>`: The name of the repository within which the witness is stored.  
`collection collection=<name>`: The name of a collection of manuscripts.  
`idno idno=<call #>`: Any form of call number.  
`msName msName=<name>`: The name commonly used for the witness.  
`origDate origDate=<date>`: Any form of date used to identify the date of origin for the witness.  
`locus locus=<locus>`: The sequence of folio references where the edition text is found in the manuscript.

*New feature v1.3*

To take here one example, a witness such as the *Marcianus Graecus* 269, referred to as manuscript ‘M’ in the editions, which contains sixty treatises transmitted under the name of Hippocrates, could be declared as follows:<sup>17</sup>—

```
\DeclareWitness{M}{M}{\emph{Marcianus Gr.} 269}[
  settlement=Venice,
  institution=Marciana Library,
  msName=Marcianus Gr.,
  idno=269,
  origDate=s. X,
  locus=fol. 416\textsuperscript{v}-426\textsuperscript{v}]
```

`\DeclareHand Hands \DeclareHand{<unique id>}{<base ms.>}{<rendition>}[<note>]`

This command requires three mandatory arguments enclosed between curly braces and one optional argument between square brackets used to specify consecutively:—

- (a) The unique identifier of the hand to be used both in the `.tex` source file and as an `xml:id` in the TEI `xml` output if any.<sup>18</sup>
- (b) The unique identifier of the witness the hand is related to. Of course, this witness must have been declared beforehand.
- (c) The rendition to be used in the printed apparatus criticus, which also will be found within the `<handNote>` element of the TEI header where the description of the hand occurs, within a `<abbr type="siglum">` element.
- (d) Some further information about the hand.

To continue the preceding example, here is how additions and corrections found in the *Marcianus Gr.* 269 could be declared after this witness has been declared itself:—

```
\DeclareHand{M1}{M}{M\textsuperscript{1}}[Emendatio scribae ipsius]
\DeclareHand{M2}{M}{M\textsuperscript{2}}[Manus posterior]
```

As can be seen, values such as M, M1 and M2 in the `.tex` source file will be printed as M, M<sup>1</sup> and M<sup>2</sup> respectively. Not only the code gains legibility, but also flexibility for simply changing any declared rendition will update corresponding sigla throughout the entire edition.

16. See <https://tei-c.org/release/doc/tei-p5-doc/en/html/MS.html#msid> for detailed information on these elements.

17. The locus specified refers to Hippocrates’ *Epidemics*, Book 6.

18. See on page 70 for more information.

As a final example, here is how ekdosis would encode information as declared above for the *Marcianus Gr.* 269 should a TEI output be required:—

```

<sourceDesc>
  <listWit>
    <witness xml:id="M">
      <abbr type="siglum">M</abbr>
      <emph>Marcianus Gr.</emph>269
    <msDesc>
      <msIdentifier>
        <settlement>Venice</settlement>
        <institution>Marciana Library</institution>
        <idno>269</idno>
        <msName>
          Marcianus Gr.
        </msName>
      </msIdentifier>
      <physDesc>
        <handDesc hands="2">
          <handNote xml:id="M1">
            <abbr type="siglum">M
            <hi rend="sup">1</hi></abbr>
            <p>Emendatio scribae ipius</p>
          </handNote>
          <handNote xml:id="M2">
            <abbr type="siglum">M
            <hi rend="sup">2</hi></abbr>
            <p>Manus posterior</p>
          </handNote>
        </handDesc>
      </physDesc>
      <history>
        <origin>
          <origDate>s. X</origDate>
        </origin>
      </history>
    </msDesc></witness>
  </listWit>
</sourceDesc>

```

`\DeclareSource Sources \DeclareSource{<unique label>}{<rendition>}`

*New feature v1.1* The *Conspectus Siglorum* that is placed ahead of the edition text is traditionally divided into two parts: a) *Codices*, which provides the list of sigla used in the apparatus, b) *Editiones uel Studia*, which provides references to sources, either published or unpublished, which contain conjectures used in the apparatus criticus. `\DeclareSource` takes two mandatory arguments used to specify consecutively:—

- (a) A unique label used in the `.tex` source file to refer to the work where the conjecture is found.
- (b) The rendition to be used in the printed apparatus criticus.



As ekdosis can include and use TEI `xml`-compliant lists of references,<sup>19</sup> it is advisable to use `Bib(L)TeX` labels in the first argument of `\DeclareSource`. Otherwise, the

19. See below [sect. 13.7 on page 76](#).

unique label used to declare the source would point to no `xml:id` and the TEI `xml` would not be valid. Likewise, shorthands fields from the bibliographical database can be recalled from within the second argument of `\DeclareSource`:—

```
\DeclareSource{Wil}{Wilamowitz}
% or for example:
\DeclareSource{Wil}{\citename{Wil}{shorteditor}}
```

`\DeclareScholar` **Scholars** `\DeclareScholar{⟨unique id⟩}{⟨rendition⟩}[⟨options⟩]`

*New feature v1.1* Occasionally, it is necessary to refer to a scholar as a person. For example, corrections and conjectures are commonly inserted as self-references to the editor of the text in the apparatus criticus in print with such words as *scripsi*, *addidi*, *correxi* and the like. Other examples come from unpublished conjectures of other scholars found in private libraries.

`\DeclareScholar` takes two mandatory arguments to specify consecutively:—

- (a) The unique identifier of the scholar to be used both in the `.tex` source file and as an `xml:id` in the TEI `xml` output if any.<sup>20</sup>
- (b) The rendition to be used in the apparatus criticus in print, which also will be found within the `<sourceDesc>` element of the TEI header where the description of the persons cited occurs, within an `<abbr type="siglum">` element.

Finally, the optional argument of `\DeclareScholar` accepts the following comma-separated list of key-value arguments:—

`rawname rawname=⟨name⟩`

`rawname` refers to a name that is not to be dissected into name part components such as forename, surname and the like. If `rawname` be used, then `ekdosis` will ignore the following three optional arguments: `forename`, `surname` and `addname`.

`forename forename=⟨forename⟩`

`forename` refers to first and middle names or initials.

`surname surname=⟨surname⟩`

`surname` stores the last name.

`addname addname=⟨additional name⟩`

`addname` refers to an additional or alternate name by which the scholar is known viz. a Latinized form of the name, a nickname, an epithet or alias.

`note note=⟨note⟩`

`note` may hold any relevant information about the material used by the editor. For example, a note may specify that this material has been found as marginal notes by the hand of the scholar in some edition in print.

`\DeclareShorthand` **Shorthands** `\DeclareShorthand{⟨unique id⟩}{⟨rendition⟩}{⟨csv list of identifiers⟩}`

This command provides a convenient way to declare *families* of witnesses. It takes three mandatory arguments used to specify consecutively:—

- (a) The unique identifier of the family to be used in the `.tex` source file.
- (b) The rendition to be used in the printed apparatus criticus.
- (c) A comma-separated list of previously declared witnesses.

As an example, the manuscripts of Caesar's *Gallic War* are divided into two families:  $\alpha$ , which includes mss. A, M, B, R, S, L and N, and  $\beta$ , which includes mss. T, f, U and l. Therefore, provided that all these witnesses have been already declared, here is how the two families  $\alpha$  and  $\beta$  could be declared:<sup>21</sup>—

20. See on page 70 for more information.

21. These witnesses are used in the example provided below in listing 7 on page 28.

```
\DeclareShorthand{a}{\alpha}{A,M,B,R,S,L,N}
\DeclareShorthand{b}{\beta}{T,f,U,l}
```

Then, symbols `a` and `b` can be used in the `.tex` source file in place of manuscripts that belong to either family.

That said, `\DeclareShorthand` is not meant to be restricted to declared witnesses. On the contrary, it also applies to any declared sources and scholars by means of `\DeclareSource` and `\DeclareScholar`. As an example, assuming that a self-reference to the person responsible for the edition has been set in the preamble, an associated shorthand can be defined like so:—

```
1 \DeclareScholar{ego}{ego}[
2   forename=John,
3   surname=Smith,
4   note=Main editor of the text]
5 \DeclareShorthand{egoscr}{\emph{scripsi}}{ego}
```

Then, the shorthand `egoscr` (l. 5) can be used to print in the apparatus criticus the technical term *scripsi* and use at the same time the pointer `#ego` that is expected in the TEI `xml` output file. Detailed examples of this technique will be provided below in [sect. 3 on page 20](#).

### 2.4.1 Printing Formatted Witnesses — Conspectus Siglorum

Once witnesses, hands, scholars and sources have been declared, `ekdosis` provides two commands to have them printed as declared from their identifiers.

`\getsiglum` `\getsiglum{⟨csv list of witnesses or single witness⟩}` behaves exactly as the `wit` optional argument of `\lem` and `\rdg` described below on pages 14 and 16. From a single identifier or from a comma-separated list of identifiers, it returns their formatted counterparts. To return to the example provided on pages 9–10, `\getsiglum{M}` would return `M`, while `\getsiglum{M1}` would return `M1`.

`\SigLine` `\SigLine{⟨unique id⟩}` returns from `⟨unique id⟩` used in the first argument of `\DeclareWitness`<sup>22</sup> a line ready to be inserted in a table set to print a Conspectus Siglorum with the following items of information separated by the symbol `&`: the siglum referring to the witness, the contents of the `description` field, followed if applicable by the sequence of folios that refers to the edition text, and the contents of the `origDate` field. An example of how one could print the Conspectus Siglorum of the manuscripts of Caesar’s *Gallic War* from the list provided on the previous page follows:—

Listing 2: Conspectus Siglorum of Caesar’s *Gallic War*

```
\begin{xltabular}[c]{0.75\linewidth}{lXl}
  \caption*{\textbf{Conspectus siglorum}}\
  \multicolumn{3}{c}{\emph{Familia} \getsiglum{a}}\
  \SigLine{A}\
  & \getsiglum{A1} \emph{Emendationes scribe ipsius} & \
  \SigLine{M}\
  [...]
  \SigLine{N}\
  \multicolumn{3}{c}{\emph{Familia} \getsiglum{b}}\
  \SigLine{T}\
  [...]
```

22. See above on page 8.

```
\SigLine{1}\\\
\end{xltabular}
```

## Conspectus siglorum

<i>Familia α</i>		
A	Bongarsianus 81	s. IX–X
	<i>A<sup>1</sup> Emendationes scribae ipsius</i>	
M	Parisinus Lat. 5056	s. XII
B	Parisinus Lat. 5763	s. IX–X
R	Vaticanus Lat. 3864	s. X
S	Laurentianus R 33	s. X
L	Londinensis Br. Mus. 10084	s. XI
N	Neapolitanus IV, c. 11	s. XII
<i>Familia β</i>		
T	Parisinus Lat. 5764	s. XI
<i>f</i>	Vindobonensis 95	s. XII
U	Vaticanus Lat. 3324	s. XI
<i>l</i>	Laurentianus Riccard. 541	s. XI–XII

## 2.5 Editing a Single Text

`ekdosis` (*env.*) Running paragraphs of one single text to be edited should be inserted in the `ekdosis` environment, like so:<sup>23</sup>—

```
\begin{ekdosis}
  Edition text goes here.
\end{ekdosis}
```

`\app` **Apparatus Entries** `\app[type=<type>]{<apparatus entries>}`

This command takes one mandatory argument and accepts one optional argument. Once references to be used as witnesses in the apparatus criticus have been declared in the preamble as described in [sect. 2.4](#) on pages 8–11, the `\app` command is used for inserting entries in the apparatus criticus, either lemmata, readings or notes, like so:—

```
I saw my friend \app{\lem{Peter}\rdg{John}} yesterday.
or:
I saw my friend
  \app{
    \lem{Peter}
    \rdg{John}
  }
yesterday.
```

`\app` accepts one further optional argument:—

`type type=<type>`


Default: default

As will be described below in [sect. 6.3](#) on page 38, `ekdosis` initially sets one layer of notes—the `default` layer—in the apparatus criticus. This layer is fit to receive notes related to variant readings from witnesses and sources used by the editor to establish

<sup>23</sup>. See above [listing 1](#) on page 5.

the edition text. Additional layers can be defined to receive other kinds of notes, such as references to texts quoted or cited in the text of the edition (*apparatus fontium*), references to testimonia, or quotations of the edited text by other authors (*apparatus testium*), explanatory notes, and so forth.<sup>24</sup> Once additional layers have been defined and assigned to new ‘types’, such as ‘testium’ and the like, these types can be used as values appended to the type ‘named option’. For more information about inserting notes in multiple-layer apparatus, see [sect. 7 on page 39](#).

**Base text and variants** As can be seen in the example above, there are two kinds of individual readings: the *lemma*, which contains the base text accepted by the editor, and the *reading*, which contains deviant readings rejected by the editor.


 What follows refers to the notions of “witness”, “source” and “scholar” as defined above on page 8.


`\lem` **Lemmata** `\lem[options]{lemma text}`

As *lemma text* is a word or a phrase judged by the editor to be authentic or authoritative, `\lem` prints it by default both in the edition text and as the first part of a new entry in the apparatus criticus, preceded by the line number where it occurs or a broad space when the entry refers to the same line as the preceding entry. The optional argument of `\lem` accepts the following comma-separated list of “name=value” arguments:—

`wit` `wit=<csv list of witnesses>`

While a single witness may be recorded as in `wit=A`, comma-separated lists of multiple witnesses must obviously be enclosed in curly braces, like so: `wit={A,B,C}`. It must be noted that witnesses can be grouped by using spaces as separators, like so: `wit={A,B,C, D,E,F}`.

 In the apparatus criticus in print, it is customary to remind the reader of the manuscript groupings by spaces or commas. ekdosis prints spaces by default, but can be instructed to print any other symbol instead.<sup>25</sup>

 Although any unique identifiers or labels used to “declare” sources and scholars as described above on pages 10–11 can also be used as values of the `wit` optional argument, it is recommended to use `sources` and `resp` to refer to either category respectively as described below.

`source` `source=<csv list of sources>`

*New feature v1.1* A “source” refers to any type of document consulted by the editor to establish the edition text. Most commonly, corrections and emendations from previous editions are cited in the apparatus criticus.<sup>26</sup>

`resp` `resp=<csv list of scholars>`

*New feature v1.1* `resp` refers to scholars responsible for the emendations, conjectures and corrections that are cited in the apparatus criticus.<sup>27</sup>

`alt` `alt=<alternate lemma>`

While the mandatory argument of `\lem`, *lemma text*, is always used to print the edition text in the upper part of the page, *alternate lemma*, if specified, supersedes what is printed in the related entry of the apparatus criticus. This mechanism is useful in more than one respect. For instance, it can be used to insert abbreviated lemmata in the apparatus criticus, or to introduce an alternate way of writing entries with Latin technical terms in the apparatus criticus as will be demonstrated below in the example provided by [listing 3](#)

24. See below, [sect. 7.2 on page 40](#).

25. See below on page 34 for details.

26. For edition texts used as sources, see examples below in [sect. 3 on page 20](#) and [sect. 13.7 on page 76](#).

27. See detailed examples in [sect. 3 on page 20](#).

on page 17.

`ilabel ilabel=<indexed label>`

*New feature v1.5* If used, `ilabel` instructs `ekdosis` not to set a label at the place where *lemma text* ends. Instead, the label is indexed as *indexed label* and only used to compute the ending line number at the place where the index is recalled by `\ilabel{<indexed label>}`.<sup>28</sup> This allows for abbreviated lemmas corresponding to spans of texts that cross verse, paragraph or section boundaries as described below in [sect. 4 on page 26](#), “*Lacunae*”.

`odelim nodelim=true|false`

*New feature v1.5* This named argument does not need a value as it defaults to `true` if used. `nodelim` removes the delimiter that is printed just before the entry in the apparatus criticus. This option is typically used in rare occasions in combination with `nolem` and `nonum` for entries that carry information not to be printed in the apparatus but nevertheless to be retained in the TEI `xml` output file.

`sep sep=<separator>`

`sep` allows to change the symbol used to separate the lemma text from deviant readings, which is by default the closing square bracket (`]`)

`nosep nosep=true|false`

This named argument does not need a value as it defaults to `true` if used. `nosep` removes the separator mentioned above. Obviously, `nosep` must be used when for some reason no `\rdg` command follows a `\lem` command that has just been used, as shown below in [listing 5 on page 24](#), l. 7.



If `nosep` be used so as to insert an explanatory note after the lemma text with the `\note` command described below on the next page, then the `sep` optional argument of `\note` can be used to put back in the separator. This technique is demonstrated below in [listing 5 on page 24](#), ll. 23–5.

`nolem nolem=true|false`

This named argument does not need a value as it defaults to `true` if used. `nolem` completely removes the lemma text from the related entry in the apparatus criticus.

`Nolem` (no-value argument)

*New feature v1.5* `Nolem` takes no value and is equivalent to `nodelim`, `nolem`, `nonum`. For an example of its usage, see [sect. 4 on page 26](#), “*Lacunae*” and [listing 6 on page 26](#).

`type type=<value>`

This named argument has no effect on the apparatus criticus of the edition in print, but it is used in the TEI `xml` output to classify the variation recorded in the entry according to some convenient typology. Categories such as lexical, morphological, orthographical and the like may apply. Obviously, `type=emendation` should be restricted to lemma texts and `type=conjecture` to variant readings recorded by means of `\rdg` described below.

`num` (no-value argument)

*New feature v1.3* `num` takes no value. If used, this argument instructs to print any line number that `ekdosis` may have decided not to print in the apparatus criticus before the lemma text.

`nonum` (no-value argument)

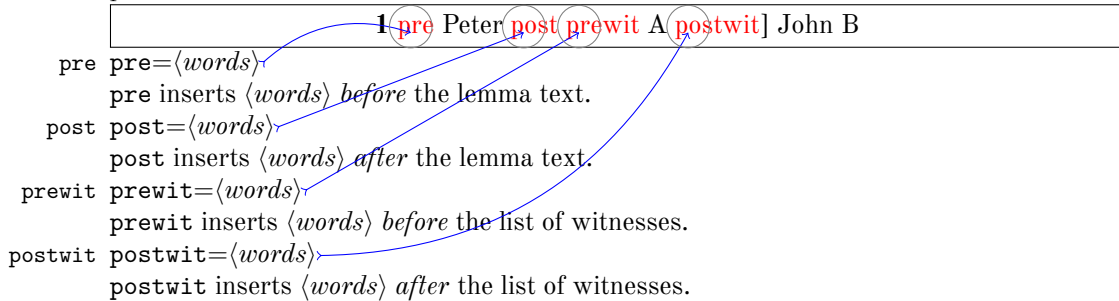
*New feature v1.3* Compared to `num`, `nonum` does the opposite. If used, any number that `ekdosis` may have decided to print before the lemma text is suppressed.

---

28. See below on this page.



Finally, four named arguments can be used to insert words at the following specific places in the lemma text:



`\rdg` **Readings** `\rdg[<options>]{<variant reading>}`

As *<reading>* is a word or a phrase judged by the editor to be unsatisfactory or corrupted, `\rdg` prints it by default in the last part of the corresponding entry in the apparatus criticus, after the symbol that is used to separate words of the base text (the lemma text) from words rejected by the editor. The optional argument of `\rdg` accepts a comma-separated list of “name=value” arguments that is almost identical to `\app`. Therefore, emphasis will be placed here only on the differences. The reader is invited to refer to the description provided above on pages 14–16 for more detailed information:—

wit `wit=<csv list of witnesses>`  
source `source=<csv list of sources>`  
resp `resp=<csv list of scholars>`  
alt `alt=<alternate reading>`  
nordg `nordg=true|false`

This named argument does not need a value as it defaults to `true` if used. `nordg` completely removes the variant reading from the related entry in the apparatus criticus.

type `type=<value>`

Obviously, `type=conjecture` should be restricted to variant readings and `type=emendation` to lemma texts recorded by means of `\lem` described above.

pre `pre=<words>`  
post `post=<words>`  
prewit `prewit=<words>`  
postwit `postwit=<words>`  
subsep `subsep=<subseparator>`

*New feature v1.4* `subsep` inserts a subseparator to be printed *before* the current entry as described below on page 36. This option is supposed to be used when no subseparator is defined, or when one is defined but for some reason a different subseparator is needed for the current entry.

`nosubsep` This argument-less option removes the subseparator from the current entry, provided one has been set by means of `\SetSubseparator`, `\SetApparatus` or `\DeclareApparatus`.<sup>29</sup>

`\note` **Notes** `\note[<options>]{<text>}` or `\note*[<options>]{<text>}`

`\note*` It may happen that editorial notes are needed to record short comments of general nature *between* lemmata and readings. `\note` inserts inline comments while `\note*` places comments below the entire apparatus block. Furthermore, if `ekdosis` be loaded with the `parnotes` option as described above on page 7, `\note*` will use the `parnotes` package to lay out the notes as an additional paragraph below the apparatus criticus. The optional argument of `\note/\note*` accepts the following comma-separated list of “name=value” arguments:—


<sup>29</sup> See on pages 36–37 and [sect. 6.3.1 on page 38](#) for details.

pre `pre=<words>`  
pre inserts *<words>* immediately before the note.

post `post=<words>`  
post inserts *<words>* immediately after the note.

sep This argument-less option is equivalent to `post=\ekdsep`.<sup>30</sup>

subsep This argument-less option is equivalent to `pre=\ekdsubsep`.<sup>30</sup>

 Under no circumstances is it permitted to insert this command `\note` or `\note*` inside the argument of `\lem` or `\rdg`. `\note/\note*` must go *between* these commands. As a general rule, within `\app{}` elements, notes are inserted immediately *after* the lemma or the variant reading they are related to. However, as will be described below in [sect. 7.2 on page 40](#), the command `\note`—with no star appended—that is used to insert explanatory notes or references to sources or testimonia is permitted within the mandatory argument of `\lem{}`, although it is subject to a very strict syntax.

[Listing 3](#) provides an illustration of some of the possibilities afforded by the commands just described:—

Listing 3: The “Peter/John” full example

```

1  \begin{ekdosisis}
2    I
3    \app{
4      \lem[wit=A]{saw}
5      \rdg[wit=B]{met}}
6    my friend
7    \app{
8      \lem{Peter}
9      \rdg{John}
10   }
11   at the station yesterday. We were both in a
12   \app{
13     \lem[wit=A]{great}
14     \rdg[wit=B]{good}}
15   mood.
16   \app{
17     \lem[wit=A, alt={How nice... said}]{\enquote{How nice to find
18       you here!} he said.}
19     \note*{There are no quotation marks in the mss.}
20     \rdg[wit=B, alt={\emph{om.}}]{}}
21   I chuckled to myself, recalling the last time we
22   \app{
23     \lem[wit=A,nolem]{met}
24     \rdg[wit=B, alt={\emph{post} met \emph{add.} there}]{met
25     there}
26     \note*{Ms. \getsiglum{B} provides other additions of this kind.}}.
27   \end{ekdosisis}

```

<sup>30</sup>. See below on page 36 for more information and [listing 5 on page 24](#), ll. 23–5 for an illustrative example.

1 I saw my friend Peter at the station yesterday. We were both in a great mood. “How  
 2 nice to find you here!” he said. I chuckled to myself, recalling the last time we met.

I saw A] met B Peter] John great A] good B 1–2 “How nice... said A]<sup>i</sup> om. B 2 post met add. there B<sup>ii</sup>

<sup>i</sup> There are no quotation marks in the mss. <sup>ii</sup> Ms. B provides other additions of this kind.

REM. 1 Close examination of lines 17–18 from [listing 3 on the preceding page](#) shows how `alt` has been used to insert an abridged lemma text in the apparatus criticus in print while keeping safe what is to be found in the TEI `xml` output.

REM. 2 The same technique has been used at line 24 to insert alternate words, including Latin technical terms, in place of the variant reading. Hence the use of `nolem` at line 23 to remove the lemma text from the apparatus criticus in print.

REM. 3 `\note*` has been used to insert short annotations in two places (ll. 19 and 26).

REM. 4 For an example of the use of `nordg`, see below [listing 7 on page 28](#), l. 11.

The corresponding TEI `xml` output produced by `ekdosis` from the L<sup>A</sup>T<sub>E</sub>X source file follows:—

Listing 4: The “Peter/John” full example: TEI `xml` output

```
<p>I
<app>
  <lem wit="#A">saw</lem>
  <rdg wit="#B">met</rdg>
</app>my friend
<app>
  <lem>Peter</lem>
  <rdg>John</rdg>
</app>at the station yesterday. We were both in a
<app>
  <lem wit="#A">great</lem>
  <rdg wit="#B">good</rdg>
</app>mood.
<app>
  <lem wit="#A">
  <quote>How nice to find you here!</quote> he said.</lem>
  <note>There are no quotation marks in the mss.</note>
  <rdg wit="#B" />
</app>I chuckled to myself, recalling the last time we
<app>
  <lem wit="#A">met</lem>
  <rdg wit="#B">met there</rdg>
  <note>Ms.
  <ref target="#B">B</ref>provides other additions of
  this kind.</note>
</app>.</p>
```


## 2.6 Indicating Subvariation in Apparatus Entries

It must be noted that grouping readings so as to keep emphasis on subvariation, regardless of its cause, is entirely optional. Furthermore, the applicability of this technique is limited to the TEI `xml` output as it helps the machines to understand a grouping otherwise immediately

accessible to human mind from the information that is available in well-written apparatus. ekdosis provides two ways of expressing subvariation.

### 2.6.1 Implicit Grouping

Because apparatus entries may nest recursively, the `\app` command can be used to group similar readings.

 However, for nesting to work, the `alt` optional argument must be used in every `\lem` and `\rdg` command involved in the nesting. This rule applies to both parent and child commands, as demonstrated in the following example:—

```
As I was walking home through Times Square, I saw my friend
\app{
  \lem[wit={A,B}, alt={Peter\---Street}]{Peter at the
    \app{
      \lem[wit=A, alt=station]{station}
      \rdg[wit=B, alt=bookstore]{bookstore}
    }
    on 42nd Street}
  \rdg[wit=C, alt={John on Broadway}]{John on Broadway}
}.
```

PDF output:—


1 As I was walking home through Times Square, I saw my friend Peter at the station on  
2 42nd Street.

I station A] bookstore B 1-2 Peter—Street AB] John on Broadway C

REM. Two `\app` commands naturally insert two entries in the apparatus criticus. As the subvariation comes first, what ms. C reads is only mentioned in the subsequent entry.

TEI xml output:—

```
<p>As I was walking home through Times Square, I saw my
friend
<app>
  <lem wit="#A #B">Peter at the
    <app>
      <lem wit="#A">station</lem>
      <rdg wit="#B">bookstore</rdg>
    </app>on 42nd Street</lem>
  <rdg wit="#C">John on Broadway</rdg>
</app>.</p>
```

 It must be noted that from a technical standpoint, albeit the nested lemmas are printed *before* their parents in the apparatus criticus, they are seen by ekdosis *after* the latter as the source file is compiled. As a result, notably when the whole nested group of lemmas falls on the same line without being preceded by an apparatus entry on this line, it may be needed to suppress redundant numbers that ekdosis may have decided to print in the apparatus criticus. Conversely, it may be needed to print numbers that ekdosis may have decided not to print. To both ends, the `num` and `nonum` optional arguments of the `\lem` command can be used as described above on page 15.

## 2.6.2 Explicit Grouping

`\rdgGrp` `\rdgGrp[(options)]{(lemma text | readings)}`  
New feature v1.1 Explicit grouping of readings can be achieved by means of the `\rdgGrp` command. It takes as mandatory argument the commands used for inserting lemma texts, readings and notes that are described on pages 14–18, viz. `\lem`, `\rdg` and `\note`. `\rdgGrp` accepts one further optional argument:—

type `type=(value)`  
This named argument is used in the TEI `xml` output to define an attribute common to all elements representing the variation.

Here follows how the technique of explicit grouping would apply to the same passage as above:—

```
As I was walking home through Times Square, I saw my friend
\app{
  \rdgGrp[type=subvariation]{
    \lem[wit=A, alt={Peter\---Street}]{Peter at the station
      on 42nd Street}
    \rdg[wit=B, alt={bookstore \emph{pro} station}]{Peter at the
      bookstore on 42nd Street}
  }
  \rdg[wit=C]{John on Broadway}
}.
```

PDF output:—

1 As I was walking home through Times Square, I saw my friend Peter at the station on  
2 42nd Street.

---

1–2 Peter—Street A] bookstore *pro* station B John on Broadway C

REM. In this example, the subvariation is emphasized with a Latin technical term and may be expressed in one single entry in a more economical manner.

TEI `xml` output:—

```
<p>As I was walking home through Times Square, I saw my
friend
<app>
  <rdgGrp type="subvariation">
    <lem wit="#A">Peter at the station on 42nd
Street</lem>
    <rdg wit="#B">Peter at the bookstore on 42nd
Street</rdg>
  </rdgGrp>
  <rdg wit="#C">John on Broadway</rdg>
</app>.</p>
```

## 3 Emendations and Conjectures

From a technical standpoint, “conjectures” are readings that are not supported by manuscript evidence, but are instead proposed by scholars to be taken into consideration for establishing the edition text. A conjecture is called an “emendation” when it is adopted

in place of what is provided by or missing from the text provided by the manuscripts. Emendations and conjectures are therefore readings and as such expected to be found within `<lem>` or `<rdg>` elements. However, as both come from editions or scholars, not from manuscripts, they are naturally associated with `source` or `resp` attributes as described above on page 14,<sup>31</sup> and can be distinguished from one another by the `type` attribute, eg. either `emendation` or `conjecture`.

As an example, the representation of witnesses, editors and shorthands of Hippocrates’ *Epidemics*, Book 2 could be summarized as follows:—

```

1  % Witnesses:
2  \DeclareWitness{V}{V}{\emph{Vaticanus Gr.} 276}
3  \DeclareWitness{I}{I}{\emph{Parisinus Gr.} 2140}
4  \DeclareHand{Iac}{I}{I\textsuperscript{ac}}[Lectio ante correctionem]
5  \DeclareHand{Ipc}{I}{I\textsuperscript{pc}}[Lectio post correctionem]
6  \DeclareWitness{R}{R}{\emph{Vaticanus Gr.} 277}
7  \DeclareWitness{H}{H}{\emph{Parisinus Gr.} 2142}
8  % Sources (the first arguments below must refer to biblalex labels and
9  % an xml bibliographical database must be supplied):
10 \DeclareSource{Lit}{Littré}
11 \DeclareSource{Erm}{Ermerins}
12 \DeclareSource{Sm}{Smith}
13 % Persons:
14 \DeclareScholar{ego}{ego}[
15     forename=Robert,
16     surname=Alessi]
17 % Useful shorthands:
18 \DeclareShorthand{codd}{codd.}{V,I,R,H}
19 \DeclareShorthand{edd}{edd.}{Lit,Erm,Sm}
20 \DeclareShorthand{egoscr}{\emph{scripsi}}{ego}

```

As can be seen from lines 18–20, three useful shorthands have been defined: `codd` prints “codd.” for Latin pl. *codices* viz. “all manuscripts” and refers to the three xml identifiers V, I, R and H declared at ll. 2–3 and 6–7; `edd` prints “edd.” for Latin pl. *editores* viz. “all editors” and refers to the three xml identifiers Lit, Erm and Sm declared at ll. 10–12;<sup>32</sup> finally, `egoscr` (l. 20) is used to print the technical Latin term “*scripsi*”, “I wrote”, to denote a personal conjecture. Then, the `.tex` source file can be structured as follows:—

```

1  \begin{ekdosis}
2  και έγινετο μάλλον \app{
3  \lem[wit={V, Ipc,R,H}]{νότω}
4  \rdg[wit=Iac]{νότου}
5  \rdg[source=Erm, type=conjecture]{έν νότω}}· [...] % conjecture
6
7  ει
8  \app{
9  \lem[resp=egoscr, type=emendation]{μέν} % emendation
10 \rdg[wit=codd, source=edd]{μη}
11 } ειη δια ταυτα [...]
12 \end{ekdosis}

```

REM. 1 Line 5 introduces a *conjecture* which has been annotated with `type=conjecture` to facilitate its identification. Other optional arguments could have been used, such as `prewit=coni.` or

31. See also on pages 10–11.

32. For detailed information on how to use `\DeclareSource` and insert references to cited works, the reader is invited to refer to [sect. 13.7 on page 76](#).

prewit=false coni., to print explanatory words in the apparatus criticus before the abridged name of the scholar.

REM. 2 Conversely, line 9 introduces an *emendation* for which the shorthand `egoscr` has been used to print the exact term *scripsi* in the apparatus criticus while keeping `ego` as an `xml:id` for the TEI `xml` output file. Other strategies could have been used. For example, one could have defined a specific shorthand to print nothing in place of `ego` and leave the insertion of technical terms to the post optional argument of `\lem`, like so:—

```
% Preamble:
% (\unskip is for removing the space left by the empty 2nd argument
% below.)
\DeclareShorthand{egomute}{\unskip}{ego}

% Document:
\app{
  \lem[resp=egomute, post=\emph{scripsi}, type=emendation]{μέν}
  \rdg[wit=codd, source=edd]{μή}
} εἴη διὰ ταῦτα [...]
```

PDF output:—

```
1 καὶ ἐγίνετο μᾶλλον νότῳ· [...]
2 εἰ μὲν εἴη διὰ ταῦτα [...]
```

---

1 νότῳ V I<sup>pc</sup>RH] νότου I<sup>ac</sup> ἐν νότῳ Ermerins 2 μὲν *scripsi*] μή codd. edd.

TEI `xml` output:—

```
<p xml:lang="grc">καὶ ἐγίνετο μᾶλλον
<app>
  <lem wit="#V #Ipc #R #H">νότῳ</lem>
  <rdg wit="#Iac">νότου</rdg>
  <rdg source="#Erm" type="conjecture">ἐν νότῳ</rdg>
</app>· [...]</p>
<p>εἰ
<app>
  <lem resp="#ego" type="emendation">μὲν</lem>
  <rdg wit="#V #I #R #H" source="#Lit #Erm #Sm">
μή</rdg>
</app>εἴη διὰ ταῦτα [...]</p>
```

### 3.1 Editorial Addition and Deletion

ekdosis provides a set of commands to indicate that text has been supplied or removed by conjecture. As regards critical symbols conventionally used for representing emendations, lacunae, omissions, gaps, editorial deletions or additions and the like, ekdosis follows the standards as described by West:<sup>33</sup>—

<> text added by conjecture or from a parallel source.

\*\*\* lacuna in the whole textual tradition.

<\*\*\*> conjectured lacuna.

{ } editorial deletion.

---

<sup>33</sup> Martin L. West, *Textual Criticism and Editorial Technique* [Applicable to Greek and Latin Texts] (Stuttgart: B. G. Teubner, 1973), 80–2.



†† text judged by the editor to be corrupt. Note that if only one word be suspect, only one crux is needed.

`\SetCritSymbols` `\SetCritSymbols{csv list of options}` can be used to change the critical symbols described above. This command accepts the following list of key-value optional arguments:—

`suppbegin` `suppbegin=symbol` Default: <

The opening symbol used to mark the text that is supplied.

`suspend` `suspend=symbol` Default: >

The closing symbol used to mark the text that is supplied.

`delbegin` `delbegin=symbol` Default: {

The opening symbol used to mark the text that is deleted.

`delend` `delend=symbol` Default: }

The closing symbol used to mark the text that is deleted.

`sicbegin` `sicbegin=symbol` Default: †

The opening symbol used to mark the text that is deemed to be suspect.

`sicend` `sicend=symbol` Default: †

The closing symbol used to mark the text that is deemed to be suspect.

`gapmark` `gapmark=symbols` Default: \*\*\*


The symbols used to mark lacunae.

`keepinapp` `keepinapp=true|false` Default: false

*New feature v1.4* This named argument does not need a value as it defaults to `true` if used. By default, the critical symbols used by `\supplied`, `\surplus` and `\sic` described below on the current page are printed in the edition text but removed from the apparatus. `keepinapp` instructs `ekdosis` to print these symbols in both places.

As an example, what follows sets `[]` for deletions and `...` for lacunae:—

```
\SetCritSymbols{
  delbegin = [,
  delend = ],
  gapmark = \dots
}
```

 If modified, brackets can be adapted to languages that are written from right to left. To that effect, `ekdosis` provides a boolean expression `al@rlmode` which is evaluated as `true` if the writing direction be set from right to left and as `false` otherwise. As the `etoolbox` package is loaded by `ekdosis`, `\ifboolean{al@rlmode}{rtl symbol}{ltr symbol}` can be used to perform the test.

`\supplied` **Editorial Addition** `\supplied{text}` is used to mark *text* that is by definition missing from the tradition as supplied by the editor or some other scholar. This command is normally expected in `\lem{}` or `\rdg{}`.

`\surplus` **Editorial Deletion** `\surplus{text}` is used to mark *text* that is deemed to be inauthentic, but nevertheless retained between braces in the edition text as it is transmitted by all witnesses. This command is normally expected in `\lem{}` or `\rdg{}`.

`\sic` **Crux** `\sic{text}` takes as mandatory argument the text deemed by the editor to be readable but not understandable. `\sic` inserts *text* between cruces while `\sic*` prints only one crux before *text*.

`\gap` **Lacuna** `\gap{<csv list of options>}` indicates that some amount of text has fallen away from the entire tradition. It takes as mandatory argument a comma-separated list of options that can be used to further specify the reason for omission, the unit of measurement, the quantity or the extent, as follows:—

`reason` `reason=<reason>`  
`reason` gives the reason for omission.

`unit` `unit=<unit>`  
`unit` provides some regularized measurement, such as `character`, `word`, `line` and the like.

`quantity` `quantity=<n>`  
`quantity` specifies the number of the given unit that comprise the measurement.

`extent` `extent=<description>`  
`extent` describes the size, including quantity and unit in a single string of words.

**Conjectured Lacuna** Assumably, the conjectured lacuna should be enclosed by `\supplied` and as such contained by `\lem` with `type=emendation` to indicate that the lacuna has been accepted by the editor.

Examples follow:<sup>34</sup>—

Listing 5: Emendations, conjectures and corrections

```

1 % Preamble:
2 \DeclareShorthand{egomute}{\unskip}{ego}
3
4 % Document:
5 \begin{ekdosis}
6   σχεδόν \app{
7     \lem[resp=egomute, nosep, post={post σχεδόν quattuor uerba
8       excidisse uid.}, type=emendation]{\supplied{\gap{reason=lost,
9         unit=word, quantity=4}}}
10  } οὗτοι
11
12  subsidiis magnis \sic*{epicuri} constabilitas
13
14  declinare quis est qui \sic{possit cernere sese}.
15
16  \app{
17    \lem[resp=egomute, type=emendation, nosep, post={ante
18      ὑπογίν.}]{\surplus{καί}}
19    \note{deleui e Gal.P}
20  } ὑπογίνονται
21
22  Πάντων δὲ \app{
23    \lem[resp=egomute, type=emendation, nosep]{\supplied{τῶν πυρετῶν}}
24    \note[sep]{addidi (\arb{^gamI`a 'l-.hummayAti}
25      \getsiglum{Gal})}
26    \rdg[nordg, source=Gal]{\arb{al-.hummayAti}}
27    \rdg[wit=codd, source=edd, alt=om.]{
28  },
29  \end{ekdosis}

```

34. On the use of `egomute` (l. 2), see above [REM. 2 on page 22](#).

PDF output:—

1 σχεδὸν <\*\*\*> οὕτοι  
2 subsidiis magnis †epicuri constabilitas  
3 declinare quis est qui †possit cernere sese†.  
4 {καὶ} ὑπογίνονται  
5 Πάντων δὲ <τῶν πυρετῶν>,

1 \*\*\* post σχεδὸν quattuor uerba excidisse uid. 4 καὶ ante ὑπογίν. deleui e Gal.P 5 τῶν πυρετῶν addidi (جميع الحُميات Gal.)]om. codd. edd.

TEI xml output:—

```
<p>σχεδὸν  
<app>  
  <lem resp="#ego" type="emendation">  
    <supplied>  
      <gap reason="lost" unit="word" quantity="4" />  
    </supplied>  
  </lem>  
</app>οὕτοι</p>  
<p>subsidiis magnis  
<sic>epicuri</sic> constabilitas</p>  
<p>declinare quis est qui  
<sic>possit cernere sese</sic>.</p>  
<p>  
<app>  
  <lem resp="#ego" type="emendation">  
    <surplus>καὶ</surplus>  
  </lem>  
  <note>deleui e Gal.P</note>  
</app>ὑπογίνονται</p>  
<p>Πάντων δὲ  
<app>  
  <lem resp="#ego" type="emendation">  
    <supplied>τῶν πυρετῶν</supplied>  
  </lem>  
  <note>addidi (  
    <foreign xml:lang="ar-Latn" type="transliterated"  
      subtype="arabtex">^gamI`a 'l-.hummayAti</foreign>  
    <ref target="#Gal">Gal.</ref></note>  
  <rdg source="#Gal">  
    <foreign xml:lang="ar-Latn" type="transliterated"  
      subtype="arabtex">al-.hummayAti</foreign>  
  </rdg>  
  <rdg wit="#V #I #R #H" source="#Lit #Erm #Sm" />  
</app>,</p>
```

## New feature v1.5 4 Lacunae

The term “lacuna” is used here to indicate a series of words or phrases that are missing from one or more witnesses in the manuscript tradition. When the lacuna is limited to just a few words, it may fill an entire entry of the apparatus criticus, especially when the other witnesses do not provide variants on this span of text.

On the other hand, when the lacuna is longer, or includes variants, or crosses a verse, paragraph or section boundary, it is usually not desirable to break it into pieces. The recommended method of encoding is therefore to record explicitly where the lacuna begins and where it ends. However, in well-written apparatus criticus, these two items of information must be provided only in the first entry. Moreover, the line numbering of this entry and the abbreviated lemma must make clear where the lacuna ends. As a consequence, the apparatus in print is silent at the place where the lacuna ends, unlike the TEI `xml` code which must be explicit at both places.

`\lacunaStart` `\lacunaStart` [*opt*] and `\lacunaEnd` [*opt*] are used in the mandatory argument of the `\rdg` command to indicate the beginning and the end of the lacuna. The optional argument of these commands can be used to specify which witnesses are affected by the lacuna, as follows:—

`wit` `wit`={*csv list of witnesses*}

This option is described above on page 14. Of course, it is unnecessary to use it in the case where only one witness is involved in the variant reading that supply the information on the lacuna.

`\ilabel` `\ilabel`{*indexed label*} is used to recall the indexed label set by means of the `ilabel` optional argument of the `\lem` command as described above on page 15. This command is used to compute the line number where the lacuna ends and must be inserted precisely at this place, supposedly outside the group formed by the `\app` command used to print the information on the lacuna in the apparatus criticus.

This technique is demonstrated by the following example taken from Cicero’s *Letters to Atticus*, VII, 9.4:<sup>35</sup>—

Listing 6: Cicero, *ad Atticum epistulae*, VII, 9.4

```
1 \begin{alignment}
2 \begin{edition}
3 Praeteriit tempus non legis, \app{
4 \lem[wit={C}, ilabel={sed7.9.4.14}, alt={sed\===legis}]{sed}
5 \rdg[wit={Ω}, alt=\emph{om.}]{\lacunaStart}
6 } libidinis tuae, fac tamen \app{
7 \lem[Nolem, wit={C}]{legis\ilabel{sed7.9.4.14}}
8 \rdg[nordg, wit={Ω}]{\lacunaEnd}}; ut succedatur
9 decernitur; impedis et ais: \enquote{habe meam rationem.}
10 \end{edition}
11 \begin{translation}
12 This term, not a legal term, but a term of your own will and
13 pleasure\===or say, this legal term\===comes to an end. The House
14 passes a decree for the appointment of a successor. You object and
15 cry, \enquote{Consider my candidature.}
16 \end{translation}
17 \end{alignment}
```

35. Cicero, *Ad Atticum epistularum libri sedecim*, recensuit H. Sjögren (Collectio scriptorum ueterum Vp-saliensis; Eranos’ Förlag, 1916), 121.13–15. English translation: Cicero, *Letters to Atticus*, ed. E.O. Winstedt (The Loeb Classical Library, 2; London – New York: William Heinemann & The MacMillan Co., 1919), 51.

- REM. 1 At line 4, from the `ilabel` optional argument, `ekdosis` knows that the span of text affected by the lacuna does not end after *sed*, but after *legis* (l. 7), at the place where the indexed label is recalled by `\ilabel`. So line numbers are computed accordingly in the apparatus criticus.
- REM. 2 The starting and ending points of the lacuna from which the  $\Omega$  family suffers have been marked by `\lacunaStart` and `\lacunaEnd` respectively (ll. 5 and 8).
- REM. 3 Line 8 shows that *legis* (l. 7) does not appear in the  $\Omega$  family. This is where the lacuna ends. Furthermore, as no other witnesses are mentioned in the `\rdg` command (l. 8), using the `wit` optional argument of `\lacunaEnd` would introduce an unnecessary redundancy.

PDF output:—

1    Praeteriit tempus non legis, sed libidinis 2    tuae, fac tamen legis; ut succedatur de- 3    cernitur; impedis et ais: “habe meam rati- 4    onem.”	This term, not a legal term, but a term of your own will and pleasure—or say, this legal term—comes to an end. The House passes a decree for the appointment of a suc- cessor. You object and cry, “Consider my candidature.”
1–2 sed—legis C] om. $\Omega$	

TEI xml output (Latin only):—

```

1 <p>Praeteriit tempus non legis,
2 <app>
3   <lem wit="#C">sed</lem>
4   <rdg wit="#Ω">
5     <lacunaStart />
6   </rdg>
7 </app>libidinis tuae, fac tamen
8 <app>
9   <lem wit="#C">legis</lem>
10  <rdg wit="#Ω">
11    <lacunaEnd />
12  </rdg>
13 </app>; ut succedatur decernitur; impedis et ais:
14 <quote>habe meam rationem.</quote></p>

```

## 5 Alignment of Parallel Texts

As already said above,<sup>36</sup> `ekdosis` can arrange sundry texts in parallel columns—synchronized or not—either on the same page or on facing pages. Depending on what is needed, any text can be equipped with an apparatus criticus. The most common example is that of an edition of a classical text with an apparatus criticus accompanied by a translation into a modern language on the facing page. One can also imagine an edition of two classical texts or two different recensions of the same text, each of which provides variants recorded in separate apparatus criticus, laid out on the left-hand pages, with one or more translations on the corresponding right-hand pages, and so forth.

`alignment` (*env.*) **The alignment Environment** `\begin{alignment} [⟨options⟩] ... \end{alignment}`

This environment can be used as it is provided to typeset a standard critical edition, namely an edition text, equipped with an apparatus criticus and laid out on the left-hand pages, accompanied by a translation into a modern language on the facing pages.

`edition` (*env.*) Within `alignment`, two environments are available by default: `\begin{edition}`

`translation (env.)` ...`\end{edition}` and `\begin{translation}... \end{translation}`. Obviously, the former is used to typeset the edition text with an apparatus criticus on the left, while the latter is used to typeset the translation on the right, like so:—

```

\begin{alignment}
  \begin{edition}
    First § of the edition text.
  \end{edition}
  \begin{translation}
    First § of the translation.
  \end{translation}
  \begin{edition}
    Second § of the edition text.
  \end{edition}
  \begin{translation}
    Second § of the translation.
  \end{translation}
\end{alignment}

```

`edition* (env.)`      Furthermore, so-called “starred” versions of these environments can be used at any point to synchronize texts, that is to print them in such a way that the tops of all paragraphs are vertically aligned. To that effect, it must be noted that merely applying this command on a single environment—for instance the leftmost one—will have all other associated paragraphs printed aligned.



While the whole edition text and the whole translation can be inserted in a single `edition/translation` environment respectively, it is recommended to enter both texts paragraph by paragraph as shown in the example above. Not only this method of encoding allows not to lose sight of paragraphs that are meant to be read together, but it is also the only way to align paragraphs in print, and it is much more suitable to mark up correspondence between spans of texts.

As an illustration, a short extract of Caesar’s *Gallic War*, VI, XIII.1 follows.<sup>37</sup> See the list of sigla for manuscripts and manuscript families above on page 11. As this document is not set for duplex printing, both texts have been put together on the same page. However, the reader will find the full `.tex` source file in [sect. 17.1 on page 90](#) and TEI `xml` output in [sect. 17.2 on page 93](#). The corresponding PDF output is available in [a separate file](#).<sup>38</sup>—

Listing 7: Caesar’s *Gallic War*, VI, 13.1

```

1  \begin{alignment}
2  \begin{edition}
3  \ekddiv{head=XIII, depth=2, n=6.13, type=section}
4  In omni Gallia eorum hominum qui \app{
5  \lem[wit=a]{aliquo}
6  \rdg[wit=b, alt=in al-]{in aliquo}}
7  sunt numero atque honore genera sunt duo. Nam plebes paene
8  seruorum habetur loco, quae \app{
9  \lem[wit={A,M}, alt={nihil audet (aut et \getsiglum{A1})}
10 per se]}{nihil audet per se}

```

36. See point (a) on page 4.

37. Latin text: Caesar, *Gallic War (Guerre des Gaules)*, ed. L.-A. Constans (Collection des Universités de France; Paris: Les Belles Lettres, 1987) (originally pub. 1926); English translation: Caesar, *Gallic War*, ed. W. A. McDevitte and W. S. Bohn (Harper’s New Classical Library; 1st edn., New York: Harper & Brothers, 1869).

38. On the use of `\ekddiv` (ll. 3 and 20), see below [sect. 11.2 on page 59](#).

```

11     \rdg[wit=A1,nordg]{nihil aut et per se}
12     \rdg[wit={R,S,L,N}]{nihil habet per se}
13     \rdg[wit=b]{per se nihil audet}}, \app{
14     \lem[wit=a]{nullo}
15     \rdg[wit=b]{nulli}} adhibetur \app{
16     \lem{consilio}
17     \rdg[wit={T, U}, alt=conc-]{concilio}}.
18 \end{edition}
19 \begin{translation}
20 \ekddiv{head=XIII, depth=2, n=6.13, type=section}
21     Throughout all Gaul there are two orders of those men who are of
22     any rank and dignity: for the commonality is held almost in the
23     condition of slaves, and dares to undertake nothing of itself,
24     and is admitted to no deliberation.
25 \end{translation}
26 \end{alignment}

```

1 XIII. In omni Gallia eorum hominum qui  
2 aliquo sunt numero atque honore genera sunt  
3 duo. Nam plebes paene seruorum habetur  
4 loco, quae nihil audet per se, nullo adhibetur  
5 consilio.

2 aliquo α] in al- β 4 nihil audet (aut et A<sup>1</sup>) per se  
AM]nihil habet per se RSLN per se nihil audet β nullo  
α] nulli β 5 consilio] conc- T U

XIII. Throughout all Gaul there are two  
orders of those men who are of any rank and  
dignity: for the commonality is held almost  
in the condition of slaves, and dares to un-  
dertake nothing of itself, and is admitted to  
no deliberation.

REM. 1 As can be seen from the apparatus entry related to l. 4 above, a subvariant has been inserted in the lemma part: “(aut et A<sup>1</sup>)”. This was done by using `alt` in [listing 7 on the preceding page](#), ll. 9–10. But as this variant is already recorded—and printed—in the lemma part, it was necessary to remove the entire otherwise redundant variant from the apparatus criticus in print. Hence the use of `nordg` at l. 11.

REM. 2 For examples of abbreviations, see ll. 6 and 17.

REM. 3 Line 17 shows how mss. T and U (which belong to two distinct subfamilies) have been separated from one another: `wit={T,U}`. See above on [page 14](#) for more information on this technique.

Finally, the corresponding TEI `xml` output follows:—

```

<div xml:id="div-edition_1" xml:lang="la">
  <div type="section" n="6.13">
    <head>XIII</head>
    <p>In omni Gallia eorum hominum qui
    <app>
      <lem wit="#A #M #B #R #S #L #N">aliquo</lem>
      <rdg wit="#T #f #U #1">in aliquo</rdg>
    </app>sunt numero atque honore genera sunt duo. Nam
    plebes paene seruorum habetur loco, quae
    <app>
      <lem wit="#A #M">nihil audet per se</lem>
      <rdg wit="#A1">nihil aut et per se</rdg>
      <rdg wit="#R #S #L #N">nihil habet per se</rdg>
      <rdg wit="#T #f #U #1">per se nihil audet</rdg>
    </app>,
    <app>
      <lem wit="#A #M #B #R #S #L #N">nullo</lem>

```



```

        <rdg wit="#T #f #U #l">nulli</rdg>
    </app>adhibetur
    <app>
        <lem>consilio</lem>
        <rdg wit="#T #U">concilio</rdg>
    </app>.</p>
</div>
</div>
<div xml:id="div-translation_1" xml:lang="en">
    <div type="section" n="6.13">
        <head>XIII</head>
        <p>Throughout all Gaul there are two orders of those men
        who are of any rank and dignity: for the commonality is
        held almost in the condition of slaves, and dares to
        undertake nothing of itself, and is admitted to no
        deliberation.</p>
    </div>
</div>

```

## 5.1 Alignment of Several Texts


As described above on page 27, the `alignment` environment may receive an optional argument in which the following “name=value” arguments are accepted:—

`tcols` `tcols`=*<number>* Default: 2  
`tcols` stores the total number of `columns` of text to be aligned.

`lcols` `lcols`=*<number>* Default: 1  
`lcols` stores the number of `columns` to be printed on the left-hand page, *out of the total number* of columns specified with `tcols`. As can be seen from the preceding two default values, `alignment` initially sets two columns of text on facing pages. Of course, for this setting to work properly, one must ensure that the `alignment` environment is started on a left page.

`texts` `texts`=*<semicolon-separated values>* Default: edition;translation  
 Depending on the total number of columns that has been specified with `tcols` above, `texts` is then used to define the names of the environments that shall receive edition texts, translations, &c. Furthermore, as described on page 28, `ekdosis` also defines “starred” versions of these environments to be used to synchronize columns so that corresponding paragraphs are printed vertically aligned. Some very important points need to be emphasized in this respect:—

- (a) Only unaccented letters of the alphabet (whatever the case) are allowed to compose the names of L<sup>A</sup>T<sub>E</sub>X environments.
- (b) These names must be separated from one another by *semicolons*, as shown in red in the listing below at the end of lines 1 and 2.

 The comma at the end of line 3 closes the whole value of `text` and acts as a higher level separator.

- (c) Each name may be followed by a ‘suboptional’ argument between square brackets which will then be used to insert TEI `xml` attributes in the corresponding `<div>` element. For example,

```

1  texts=latin[xml:lang="la"];
2      english[xml:lang="en"];
3      french[xml:lang="fr"],

```

will be converted into TEI xml as follows:—

```
<div xml:id="div-latin_1" xml:lang="la">
...
</div>
<div xml:id="div-english_1" xml:lang="en">
...
</div>
<div xml:id="div-french_1" xml:lang="fr">
...
</div>
```



As can be seen, `ekdosis` takes care of computing and inserting the `xml:id` attributes which are therefore not accepted in the ‘suboptional’ arguments of `texts`.

- (d) The names of the environments must be specified in exactly the same order as they are supposed to appear in the print edition, from left to right.

- `apparatus` `apparatus=(semicolon-separated values)` Default: edition  
Then, the `apparatus` option, just as `texts`, takes a semicolon-separated list of previously defined environments that shall receive at least one layer of apparatus criticus.
- `paired` `paired=true|false` Default: true (initially not set)  
This named argument does not need a value as it defaults to `true` if used. By default, `ekdosis` follows the L<sup>A</sup>T<sub>E</sub>X page numbering scheme when multiple texts are arranged on facing pages. The `paired` option leaves every right-hand page number unchanged, so that both facing pages hold the same page number.<sup>39</sup>
- `lineation` `lineation=page|document` Default: document  
This option applies to edition texts initially set to receive an apparatus criticus. By default, lines are continuously numbered throughout the document. `lineation=page` sets the numbering to start afresh at the top of each page.
- `flush` `flush=true|false` Default: false  
This named argument does not need a value as it defaults to `true` if used. This option applies when two or more distinct `alignment` environments are started on the same page. Should this happen, any subsequent `alignment` environment must be set with the `flush` option so that every one of them carry its own apparatus criticus.

As an example, the alignment of the Latin edition text of Caesar’s *Gallie War*, printed on left-hand pages, along with two translations into English and French, printed on right-hand pages, can be set as follows:—

```
\begin{alignment}[tcols=3,
                 lcols=1,
                 texts=latin[xml:lang="la"];
                 english[xml:lang="en"];
                 french[xml:lang="fr"],
                 apparatus=latin,
                 lineation=page]

\begin{latin}
Gallia est omnis divisa in partes tres quarum unam incolunt
Belgae, [...]
\end{latin}
\begin{english}
```

<sup>39</sup>. For setting headers and footers in this arrangement of text, but also for the current limitations attached to this option, see on page 63.

```

All Gaul is divided into three parts, one of which the Belgae
inhabit, [...]
\end{english}
\begin{french}
L'ensemble de la Gaule est divisé en trois parties: l'une est
habitée par les Belges, [...]
\end{french}
\end{alignment}

```

`\SetAlignment` `\SetAlignment{alignment settings}`  
If the same alignment settings be shared by several `alignment` environments, common settings can be collected in the argument of `\SetAlignment`, like so:—

```

\SetAlignment{
  tcols=3,
  lcols=1,
  texts=latin[xml:lang="la"];
  english[xml:lang="en"];
  french[xml:lang="fr"],
  apparatus=latin,
  lineation=page
}
\begin{alignment}
...
\end{alignment}

```

`\SetAlignment` can be used either in the preamble or at any point of the document to set or to modify alignment settings.

### 5.1.1 Appending Hooks to Environments

`\AtBeginEnvironment` Once environments corresponding to texts to be aligned have been defined, it is advisable to use the `\AtBeginEnvironment{environment}{code}` command to further adjust languages, hyphenation rules, and/or fonts to be applied in each environment. To return to the example provided above, once `\SetAlignment` has been used, the languages can be set as follows:—


```

\AtBeginEnvironment{latin}{\selectlanguage{latin}}
\AtBeginEnvironment{english}{\selectlanguage{english}}
\AtBeginEnvironment{french}{\selectlanguage{french}}

```

## 5.2 Laying Out Parallel Texts

As `ekdosis` uses the `paracol` package for the layout of parallel texts, most of the commands provided by this package apply. In this respect, quite useful are the commands described in sections 7.3 to 7.6 on pp. 15–21 of the documentation of this package.<sup>40</sup>

 It must be noted that all these commands are to be inserted *before* the `alignment` environments on which they are supposed to operate.

40. Hiroshi Nakashima, *The Paracol package* (version 1.35) [Multiple columns with texts “in parallel”] (Dec. 31, 2018), <http://www.ctan.org/pkg/paracol>.

### 5.2.1 Columns and Gutters

`\columnratio` **Column Ratio on Single Pages** `\columnratio{\langle r_1, r_2, \dots, r_n \rangle}`, where  $r_1$  refers to the leftmost column, can be used to set the ratio of the columns in relation to each other. Depending on the total number of columns on which one wishes to operate, a comma-separated list of decimal numbers is expected. As an example, `\columnratio{0.6}` will instruct `ekdosis` to have the first column spread over 60 % of the total width of the text block, minus the total width of intercolumnar gutters.

**Column Ratio on Facing Pages** `\columnratio` accepts an optional argument which can be used as described above to set the ratio of columns to be printed on right-hand pages, like so: `\columnratio{\langle r_1, r_2, \dots, r_n \rangle}[\langle r_1, r_2, \dots, r_n \rangle]`.

`\setcolumnwidth` **Column Width on Single Pages** `\setcolumnwidth{\langle w_1, w_2, \dots, w_n \rangle}` operates the same way as `\columnratio` described above, except that dimensions are expected instead of ratios. As an example, `\setcolumnwidth{1in}` will have the width of the first column set to 1 in.

**Gutter Width** Each value accepted by `\setcolumnwidth` can be expressed as a pair as in `\setcolumnwidth{\langle w_1/g_1, w_2/g_2, \dots, w_n/g_n \rangle}` where the character `/` acts as a separator, in which case  $g_x$  is used to set the width of the gutter that follows the  $x^{\text{th}}$  column. As an example, `\setcolumnwidth{1in/0.25in}` will print a 1 in first column, followed by a 0.25 in gutter.

**Automatically Computed Values** Widths of columns and widths of gutters can be replaced with `\fill` and `\columnsep` respectively. As an example, `\setcolumnwidth{\fill/0.25in}` will only operate on the width that follows the first column, all remaining values being computed automatically.

**Column and Gutter Width on Facing Pages** Just like `\columnratio`, `\setcolumnwidth` accepts an optional argument which can be used to set the width of columns and gutters to be printed on right-hand pages, like so: `\setcolumnwidth{\langle w_1, w_2, \dots, w_n \rangle}[\langle w_1, w_2, \dots, w_n \rangle]` for columns only, and `\setcolumnwidth{\langle w_1/g_1, w_2/g_2, \dots, w_n/g_n \rangle}[\langle w_1/g_1, w_2/g_2, \dots, w_n/g_n \rangle]` for columns and gutters.

**Vertical Rules** Vertical rules between columns can be drawn by setting the length of the L<sup>A</sup>T<sub>E</sub>X `\columnseprule` register to a non-zero value, like so:—

```
\setlength{\columnseprule}{0.4pt}
```

### 5.2.2 Marginal Notes

By default, marginal notes that refer to the first column are printed in the left margin, while notes that refer to subsequent columns are printed in the right margin.

`\marginparthreshold` `\marginparthreshold{\langle n \rangle}`, where  $n$  is an integer, can be used to change the default settings. This command instructs `ekdosis` that columns of text, up to the  $n^{\text{th}}$  column included, shall have their marginal notes printed to the left. As a result, to take an example, `\marginparthreshold{0}` will have all marginal notes printed in the right margin. `\marginparthreshold` also accepts an optional argument, namely `\marginparthreshold{\langle n \rangle}[\langle n' \rangle]`, that can be used to set the threshold for columns printed in right-hand pages.

### 5.2.3 Regular Footnotes

`\footnotelayout` By default, regular footnotes are printed at the bottom of the column on which they are called. `\footnotelayout{⟨key-letter⟩}` can be used to change this setting. This command accepts as mandatory argument a key-letter which can be either `c`, `p` or `m`. `c` means *column-wise* footnotes, which is the default value. `p` means *page-wise*: footnotes from all columns are gathered in a single spanning block at the bottom of the page. Finally, `m` stands for *merge*, which means that all footnotes that are called on a given page, including notes that are called outside the `alignment` environment, are printed in a single spanning block at the bottom of the page.



Regular footnotes are printed above the block of critical notes. The respective places of these blocks can be interchanged by just loading the `fnpos` package in the preamble.<sup>41</sup>

## 6 Laying Out the Apparatus Criticus

### 6.1 General Hooks

Some hooks are shared by all layers of notes that are inserted in the apparatus criticus (e.g. sources, testimonia, variant readings &c.)

`\SetHooks` `\SetHooks{⟨csv list of hooks⟩}` can be used either in the preamble or at any point of the document. The list of accepted hooks at the time of writing follows:—

`appfontsize` `appfontsize=⟨command⟩` Default: `\footnotesize`  
This option sets the size of the font to be used in the whole apparatus criticus. By default, it is the same as the size used for footnotes.

`refnumstyle` `refnumstyle=⟨command⟩` Default: `\bfseries`  
`refnumstyle` can be used to set the family, series or shape of the font used to print references to line numbers in the apparatus criticus. By default, numbers are printed in bold face. As an example, `refnumstyle=\normalfont` will have them printed in the font and shape selected by default for the document, while `refnumstyle=\bfseries\itshape` will have them printed in bold and italic.

`postrefnum` `postrefnum=⟨command | chars⟩` Default: `~`  
`postrefnum` can be used to set what immediately follows the reference to line numbers. By default, it is `~`, namely an unbreakable space. As an example, `postrefnum=\hskip 0.5em` will insert a 0.5 em space between the numerals and the beginning of all subsequent notes.

`lemmastyle` `lemmastyle=⟨command⟩` Default: `not set`  
*New feature v1.2* `lemmastyle` can be used to set the family, series or shape of the font used to print lemma texts in the apparatus criticus. For example, `lemmastyle=\bfseries` will print all instances of lemma text in bold while `lemmastyle=\color{blue}` will print them in blue.

`readingstyle` `readingstyle=⟨command⟩` Default: `not set`  
*New feature v1.2* `readingstyle` operates in the same way as `lemmastyle` but applies to variant readings.

`familysep` `familysep=⟨symbol⟩|reset` Default: `not set`  
*New feature v1.4* As described above on page 14, multiple witnesses must be recorded in the `wit` optional argument of the `\lem` and `\rdg` commands as comma-separated lists of sigla which can be further grouped into families by inserting spaces as separators just after commas at specific places. `ekdosis` saves and prints these spaces in the apparatus criticus but can be instructed to print any other symbol instead by means of `familysep`. For instance, `familysep={,}` has the separating spaces replaced with commas and `familysep={,\allowbreak}` does the same while further allowing breaks after commas at the end of lines. Finally,

---

41. Hiroshi Nakashima, *The Fnpos package* (version 1.0) [Control the position of footnotes on the page] (Sept. 3, 2018), <http://www.ctan.org/pkg/fnpos>.

familysep=reset can be used to restore the default behavior.

`initialrule` `initialrule=<command>` Default: `\rule{0.4\columnwidth}{0.4pt}` (initially not set)

*New feature v1.2* `initialrule` draws a separating rule between the edition text and all subsequent layers of critical notes. Of course, this option only makes sense when multiple-layer apparatus criticus are set. Therefore, it has no effect on single-layer apparatus criticus.

`noinitialrule` Default: not set


This is a no-value option. It removes any previously set `initialrule`.

An example of the way these hooks may be used is provided below in [listing 8 on page 48](#).

### Limiting the Number of Entries per Page

`maxentries` `maxentries=<n>` (where  $n \geq 10$ ) Default: not set

*New feature v1.5* This option may be used to set a limit to the number of entries per page that *all layers of critical notes taken together* may accept, notably when entries are so abundant in number that `ekdosis` may oscillate indefinitely between different sets of page decisions without being able to settle down.

 It is also possible to restrict this option to a given layer of the apparatus criticus as described below on [page 39](#). If `maxentries` be set both globally in `\SetHooks` and in `\DeclareApparatus` for specific layers, then the page will break when *any* of the values defined as `maxentries` has been reached.

If `maxentries=<n>` be set, then `ekdosis` will issue `\pagebreak` (namely `\penalty-10000`) just after the  $n^{\text{th}}$  entry has been inserted in whichever layer of the apparatus criticus. As a result, the page will actually break at the end of the current line. The particulars of this technique will be discussed below in [sect. 12.1 on page 65](#).

`nomaxentries` Default: not set

`nomaxentries` does not accept any value and is equivalent to `maxentries=none`.

### Option Specific to the `layout=keyfloat` Global Setting<sup>42</sup>

`keyparopts` `keyparopts=<csv options>` Default: empty

*New feature v1.3* The comma-separated options that can be used are those described in the documentation of the `keyfloat` package.<sup>43</sup> As an example, `keyparopts={ft, tr={made with ekdosis}, lw=1.2}` will draw a tight frame around the apparatus block, have the words “made with `ekdosis`” printed below this block on the right and set its width to `1.2\linewidth`.

### Options Specific to the `layout=fitapp` Global Setting<sup>44</sup>

`appheight` `appheight=<dimension>` Default: `0.5\textheight`

*New feature v1.3* This option is used to change the maximum height up to which the apparatus block is allowed to grow before the size of the characters is reduced to allow for more entries. The value must be a dimension, namely a number followed by a length unit, such as `0.65\textheight`, `18cm` or `6in`.

`fitalgorithm` `fitalgorithm=fontsize|hybrid|areaseize|squeeze` Default: `fontsize`

*New feature v1.3* The four algorithms that can be used to have the entries inserted in the apparatus criticus fit to the selected height are presented here from the tightest to the loosest, that is, the slowest to the fastest.<sup>45</sup> While it is advisable to limit the use of `fontsize` to high quality typesetting for camera-ready copies, `areaseize` offers a satisfactory settlement when speed must be given an advantage for intermediate or draft copies. `squeeze` should be avoided as it gives results that are offensive to the sight and unacceptable to any reader.

42. See above [\(c\) on page 6](#).

43. Dunn (cf. n. 5), sect. 2.3, pp. 13–5.

44. See above [\(d\) on page 6](#).


45. See Sturm (cf. n. 8), 446–9 for details and illustrative examples.

## 6.2 Single-Layer Apparatus Criticus


**Specific Commands** Single-layer apparatus criticus can be laid out in a variety of ways with the following specialized commands, all of which can be used in the preamble or at any point of the document:—

`\SetLTRapp`     `\SetLTRapp` and `\SetRTLapp` are two argument-less commands to set the direction of the apparatus criticus, either left-to-right or right-to-left.  
`\SetRTLapp`  
`\SetSeparator`     `\SetSeparator{<separator>}` is used to change the separator between lemma texts and variant readings. By default, the separator is a closing square bracket followed by a space (]␣).

`\SetSubseparator`     `\SetSubseparator{<subseparator>}` is used to set or change the “subseparator” between succeeding variant readings. By default, no subseparator is set.  
*New feature v1.4*

 As the subseparator applies to subsequent variant readings only, it is naturally preceded by a breakable space. This space can be removed by `\unskip`. As an example, what follows replaces the breakable space with an unbreakable space, then prints a colon as subseparator followed by a space:—

```
\SetSubseparator{\unskip~: }
```

`\ekdsep`      Once the separator and if applicable the subseparator have been set, they can be accessed by `\ekdsep` and `\ekdsubsep` respectively. It is therefore advisable to use these commands instead of the mere symbols at whatever place one would have them printed.  
`\ekdsubsep`

`\SetBeginApparatus`     `\SetBeginApparatus{<characters|commands>}` can be used to append `<characters>` or `<commands>` at the beginning of the apparatus block. By default, nothing is appended. For instance, `\SetBeginApparatus{\textbf{Apparatus:}}` will append “**Apparatus:**” at the beginning of the apparatus block, while `\SetBeginApparatus{\hskip 1em}` will set an indentation of one em.

`\SetEndApparatus`     `\SetEndApparatus{<characters>}` can be used to append `<characters>` at the end of the apparatus block. By default, nothing is appended. As an example of use, `\SetEndApparatus{.}` will have a period printed at the end of the apparatus as it is customary in some editions.<sup>46</sup>

`\SetUnitDelimiter`     `\SetUnitDelimiter{<delimiter>}` can be used to set the delimiter between entries in the apparatus criticus. By default, there is no delimiter except a simple space. `<delimiter>` can be a broad space (such as `\hskip 0.75em` for instance as in the OCT series) or the divider-sign (||, as in the Budé series).

`\SetDefaultRule`     By default, `ekdosis` draws a separating line between the edition text and the apparatus criticus. This line is initially defined as `\rule{0.4\columnwidth}{0.4pt}`. `\SetDefaultRule{<line definition>}` can be used in the preamble or at any point of the document to change the default setting. Leaving this argument empty as in `\SetDefaultRule{}` removes the line.

`\SetApparatusLanguage`     `\SetApparatusLanguage{<language name>}` can be used when it is needed to apply in the apparatus criticus a language different from the one that is selected in the edition text.  
*New feature v1.2*

`\SetApparatusNoteLanguage`     `\SetApparatusNoteLanguage{<language name>}` can be used when it is needed to apply in text entries introduced by the mandatory argument of the `\note` command as described in [sect. 7.2 on page 40](#)—namely `\note[<options>]{<text>}`—a language different from the one that is selected in the edition text.  
*New feature v1.3*

`\SetApparatus`     **General Command**     `\SetApparatus{<csv list of apparatus settings>}`

46. See also below on page 66 on how to remove superfluous dots.



Finally, all the settings described above can also be collected in the argument of `\SetApparatus`. `\SetApparatus` accepts the following list of comma-separated key=value options:—

<code>direction</code>	<code>direction=LR RL</code>	Default: LR
	The writing direction of the apparatus criticus, either left-to-right (LR) or right-to-left (RL).	
<code>sep</code>	<code>sep=&lt;command   chars&gt;</code>	Default: ]␣
	The separator between lemma texts and variant readings.	
<code>subsep</code>	<code>subsep=&lt;command   chars&gt;</code>	Default: not set
	The “subseparator” between succeeding variant readings.	
<code>delim</code>	<code>delim=&lt;delimiter&gt;</code>	Default: not set
	The delimiter between entries in the apparatus criticus. As said above, there is no default delimiter except a simple space.	
<code>bhook</code>	<code>bhook=&lt;characters commands&gt;</code>	Default: empty
	The characters or commands to be appended at the beginning of the apparatus block.	
<code>ehook</code>	<code>ehook=&lt;characters&gt;</code>	Default: empty
	The characters to be appended at the end of the apparatus block. <sup>47</sup>	
<code>rule</code>	<code>rule=&lt;command&gt;</code>	Default: <code>\rule{0.4\columnwidth}{0.4pt}</code>
	As described above, <code>rule</code> is used to draw the separating line between the edition text and the apparatus criticus.	
<code>norule</code>		Default: not set
	<code>norule</code> does not accept any value and is used to remove the line.	
<code>lang</code>	<code>lang=&lt;language name&gt;</code>	Default: not set
<i>New feature v1.2</i>	<code>lang=&lt;language name&gt;</code> is used as described on the preceding page when it is needed to apply in the apparatus criticus a language different from the one that is selected in the edition text. <code>language name</code> can be any value accepted by <code>babel</code> or <code>polyglossia</code> .	
<code>notelang</code>	<code>notelang=&lt;language name&gt;</code>	Default: not set
<i>New feature v1.3</i>	<code>notelang=&lt;language name&gt;</code> is used as described on the previous page when is needed to apply in text entries introduced by the mandatory argument of the <code>\note</code> command as described in <a href="#">sect. 7.2 on page 40</a> —namely <code>\note[&lt;options&gt;]{&lt;text&gt;}</code> —a language different from the one that is selected in the edition text. <code>language name</code> can be any value accepted by <code>babel</code> or <code>polyglossia</code> .	

As an example, an apparatus criticus with references to line numbers printed in normal font, a colon as a separator between lemma texts and variant readings, a broad space as a delimiter between entries and a 0.7 in line above could be laid out as follows:—

```
\SetHooks{
  refnumstyle=\normalfont
}
\SetApparatus{
  sep={: },
  delim=\hskip 1em,
  rule=\rule{0.7in}{0.4pt}
}
```

`\footnoteruletrue` **Footnote Separator** As already seen above, `ekdosis` takes care of drawing a separating line between the edition text and the apparatus criticus. Therefore, it may be not desirable to have the standard L<sup>A</sup>T<sub>E</sub>X “`\footnoterule`” printed on every page where regular footnotes are


47. See also n. 46 on the previous page.



found. `\footnoterulefalse` removes it while `\footnoteruletrue` leaves it untouched. The latter is set by default.

## 6.3 Multiple-Layer Apparatus Criticus

As said above in (b) on page 4, `ekdosis` can print edition texts equipped with multiple-layer apparatus criticus. To take an example, most classical editions provide at least two layers of notes: one to collect references to testimonia or parallel passages (`apparatus testium`) and the other to record variant readings (the apparatus criticus *stricto sensu*). The former is always printed above the latter.

 The default single-layer apparatus criticus that is described above in sect. 6.2 on page 36 is called `default` internally. If any additional layer of notes be declared in the preamble, this `default` layer must be included in the list of declared layers.

`\SetDefaultApparatus` `\SetDefaultApparatus{<name>}` can be used at any point of the document to change the name to be used as the default one by `ekdosis`.

### 6.3.1 Declaring Additional Layers

`\DeclareApparatus` `\DeclareApparatus{<name>}[<csv list of apparatus settings>]` is a preamble-only command. As a mandatory argument, it takes the name of the new layer of notes to be inserted in the apparatus block. Declared layers are then printed one below the other in the exact same order as they are declared in the preamble. Therefore, one additional layer meant to print the testimonia above the variant readings (`apparatus testium`) can be declared as follows:—

```
1 % preamble:
2 \DeclareApparatus{testium}
3 \DeclareApparatus{default}
```

In this example, `testium` is a new name for `default`, as said just above, is already known to `ekdosis` and used as the default layer of notes. Furthermore, as `testium` is declared before `default`, `ekdosis` will print the testimonia at the top of the apparatus block.

### 6.3.2 Laying Out Layers With The Optional Argument of `\DeclareApparatus`

`direction` With regard to layout, any declared layer inherits the default values described above in sect. 6.2 on page 36. That said, as the optional argument of `\DeclareApparatus` accepts the exact same key-value options as `\SetApparatus` described on pages 36–37, `ekdosis` provides a straightforward mechanism to have any layer printed in a distinct layout.

`bhook` To return to the example provided on the preceding page, one could keep the same settings as above for the variant readings, declare an apparatus `testium` with a closing `ehook` square bracket as a separator and finally remove the line between the testimonia and the `norule` variant readings like so:—

```
lang \SetHooks{ 1
notelang   refnumstyle=\normalfont, 2
           initialrule=\rule{0.7in}{0.4pt} 3
           } 4
\DeclareApparatus{testium}[ 5
  sep={] }, 6
  delim=\hskip 1em, 7
  norule 8
```

```


]                               9
\DeclareApparatus{default}[    10
  sep={: },                     11
  delim=\hskip 1em,            12
  rule=\relax                   13
]                               14

```

- REM. 1 The general hook `initialrule` used here (l. 3) is described above on page 35.
- REM. 2 `\relax` (l. 13) is a TeX primitive that instructs to do nothing. Therefore, `rule=\relax` is not strictly equivalent to `norule`: with the former, `\relax` removes the rule but leaves untouched the subsequent carriage return: as a result, the layers are visually separated from one another by a blank line. With the latter everything is removed, carriage return included.

**Limiting the Number of Entries per Page** In some instances, it may be useful to set a limit to the number of entries per page that a given layer of critical notes may accept, notably when entries are so abundant in number that `ekdosis` may oscillate indefinitely between different sets of page decisions without being able to settle down.

`maxentries` `maxentries=<n>` (where  $n \geq 10$ ) Default: not set

 It is also possible to set a maximum number of entries for all layers of critical notes taken together as described above on page 35. If `maxentries` be set both globally in `\SetHooks` and in `\DeclareApparatus` for specific layers, then the page will break when *any* of the values defined as `maxentries` has been reached.

If `maxentries=<n>` be set, then `ekdosis` will issue `\pagebreak` (namely `\penalty-10000`) just after the  $n^{\text{th}}$  entry has been inserted in the layer of the apparatus criticus this option is related to. As a result, the page will actually break at the end of the current line. The particulars of this technique will be discussed below in [sect. 12.1 on page 65](#).

## 7 Inserting Notes in Multiple-Layer Apparatus

As said above in [sect. 6.3 on the preceding page](#), `ekdosis` initially sets one layer of notes that is called the “default” layer. As a result, any note inserted within the argument of `\app{}` as described on page 13 will go into that layer of the apparatus, unless `\SetDefaultApparatus` has been used to set another name for the default layer (see above on the preceding page).

### 7.1 Variant Readings

In most cases, all variant readings go into the “default” layer of the apparatus criticus. But in some other cases, for example when the manuscripts used refer to different recensions, it may happen that one wishes to record the related variants in separate layers. As already described on page 13, the `type` optional argument of the `\app` command can be used to insert lemma texts and associated variants in any other ‘declared’ layer of the apparatus criticus.

The following example assumes that some edition text is received in two different recensions and the variant readings that belong to the first recension are recorded in the default layer of notes while those of the second recension are to be printed in a second layer, below the default one. First, both layers must be declared in the preamble in sequence, like so:—

```

\DeclareApparatus{default} % default layer
\DeclareApparatus{rec2} % additional layer below the default one

```


Should one wish to refer to `rec1` as the default layer, then `\SetDefaultApparatus` must be used, like so:—

```
\SetDefaultApparatus{rec1}
\DeclareApparatus{rec1} % new layer set as default
\DeclareApparatus{rec2} % additional layer below the default one
```

Then, whatever option has been chosen, lemma texts and variants inserted with `\app{}` will go into the upper, default layer of notes, while those inserted with `\app[type=rec2]{}`  will go into the lower one:—


```
Some \app{
  \lem{word}
  \rdg{reading}
} to go into the default layer of notes.


Some \app[type=rec2]{
  \lem{note}
  \rdg{comment}
} to be recorded as part of the second recension.
```

 At any rate, `type=default` or `type=rec1`, depending on what has been chosen, must be used should the editor wish to retain that information in the TEI `xml` output file.

## 7.2 Other Notes for Comments, Sources or Testimonia

Additional layers of notes can be used to print short comments or to record references to texts quoted by the author of the edited text or references to the edited text by other authors or translators. The former set is called an *apparatus fontium* while the latter is called an *apparatus testimonia*.

 From a technical standpoint, these notes are very different from the short editorial notes inserted between lemma texts and variant readings that have been described above on page 16. However, for the sake of consistency with TEI `xml` encoding, `ekdosis` uses the same command `\note` to insert both kinds of notes.

 One must also keep in mind that the notes that are described in this section refer either to a single word or to a span of text. By consequence, as boundaries must always be set outside spans of text, notes must be inserted immediately before the word or words they are related to. As a result of this rule, all spaces subsequent to `\note` are ignored.

`\note` `\note[options]{text}`

As said above, `\note`, when found outside `\app{}`, is used to insert in additional layers of the apparatus short comments or references to texts quoted or cited in the edition text. It accepts the following comma-separated list of `key-value` optional arguments:—

`type type=<type>`

`type` is used to specify the name of the layer where the note is to be printed.<sup>48</sup>

`nodelim nodelim=true|false`

*New feature v1.5* This named argument does not need a value as it defaults to `true` if used. This option removes the delimiter that is printed just before the note in the apparatus criticus.

`sep sep=<command | chars>`

The separator between the lemma text and the contents of the note.

`nosep nosep=true|false`

<sup>48</sup>. See [sect. 6.3.1 on page 38](#) to learn how to declare and lay out new layers of notes.

This named argument does not need a value as it defaults to `true` if used. Obviously, `nosep` removes the separator mentioned above.

`lem` `lem=<lemma text>`

`lem` is the span of text the note is about. It may consist of one or more words, or of an abridged lemma text.

`num` (no-value argument)

*New feature v1.3* `num` takes no value. If used, this argument instructs to print any line number that `ekdosis` may have decided not to print in the apparatus criticus before the note.


`nonum` (no-value argument)

Compared to `num`, `nonum` does the opposite. If used, any number that `ekdosis` may have decided to print before the note is suppressed.

`labelb` `labelb=<label>`

Mandatory

`labelb` is the unique label to serve as a reference for the point immediately preceding the lemma text.

 `labelb` is used by `ekdosis` to print the line numbers in the apparatus criticus and to set the `left()` XPointer should TEI output be required. Therefore, it must be specified. Otherwise, `ekdosis` will issue an error message. However, two strictly consecutive `\note` commands are allowed to share the same `labelb` value for it may happen that consecutive notes need to refer to spans of text that begin at the exact same location. In this case, `ekdosis` generates only one `\lineLabel` and one corresponding `<anchor>` element in the TEI xml file.

`labelc` `labelc=<label>`

`labelc` is the unique label to serve as a reference for the point immediately following the lemma text. Contrary to `labelb`, `labelc` may be left unspecified if the note be only about one word. If the note be about a span, then `labelc` must be specified.

`\lineLabel` `\lineLabel{<label>}`

If `labelc=<some_label>` be specified in the optional argument of `\note`, `\lineLabel{<some_label>}` must be inserted immediately after the span of text that the note is about so that `ekdosis` can locate the exact point where the lemma text addressed by the note ends, like so:—

```
% Preamble:
% \DeclareApparatus{fontium}[
%     delim=\hskip0.75em,
%     bhook=\textbf{Sources:},
%     ehook=.]
% \DeclareApparatus{default}[
%     delim=\hskip0.75em,
%     ehook=.]
% Document:
\begin{ekdosis}
  The oldest monument of the Germans is their language, which, before
  untold centuries, was the companion of their travels from central
  Asia; a language, copious, elastic, inviting self-explaining
  combinations and independent development; lending itself alike to
  daily life and imagination, to description and abstract thought.
  \note[type=fontium, labelb=B61e, labelc=B62a, lem={They
    had... slave}]{Waitz, \emph{Deutsche Verfassungs Geschichte},
    i. 86} They had a class of nobles, but their tongue knew no word
    for slave.\lineLabel{B62a}\footnote{George Bancroft, \emph{History
    of the United States from the Discovery of the American
```

```
Continent}, II.61--2.}
\end{ekdosis}
```

PDF output:—

1 The oldest monument of the Germans is their language, which, before untold centuries,  
2 was the companion of their travels from central Asia; a language, copious, elastic, inviting  
3 self-explaining combinations and independent development; lending itself alike to daily life  
4 and imagination, to description and abstract thought. They had a class of nobles, but their  
5 tongue knew no word for slave.<sup>49</sup>

---

Sources: 4–5 They had... slave] Waitz, *Deutsche Verfassungs Geschichte*, i. 86

TEI xml output:—

```
<p>The oldest monument of the Germans is their language,
which, before untold centuries, was the companion of their
travels from central Asia; a language, copious, elastic,
inviting self-explaining combinations and independent
development; lending itself alike to daily life and
imagination, to description and abstract thought.
<note type="fontium" target="#range(right(B61e),left(B62a))">Waitz,
<emph>Deutsche Verfassungs Geschichte</emph>, i. 86</note>
<anchor xml:id="B61e" />They had a class of nobles, but
their tongue knew no word for slave.
<anchor xml:id="B62a" />
<note place="bottom">George Bancroft,
<emph>History of the United States from the Discovery of
the American Continent</emph>, II.61--2.</note></p>
```

`\note` or `\linelabel` inside `\lem` It may happen that the `\note` or `\linelabel` command is found inside the argument of `\lem`. Obviously, inserting such a command in the apparatus criticus in print makes no sense and will lead to an error. The solution is to insert in the value of the `alt` optional argument of `\lem` a duplicate of the lemma text devoid of that command, like so:—

```
This is some \app{
  \lem[alt=dummy]{\note[type=fontium, labelb=bnote, labelc=enote,
    lem=dummy... command]{Text of the note.}
  dummy}
  \rdg{pseudo}}
text to demonstrate how to insert a note in the argument of the
\emph{lem} command.\linelabel{enote}
```

PDF output:—

---

49. George Bancroft, *History of the United States from the Discovery of the American Continent*, II.61–2.

1 This is some dummy text to demonstrate how to insert a note in the argument of the  
2 *lem* command.

---

Sources: 1–2 dummy... command] Text of the note.

---

1 dummy] pseudo

TEI xml output:—

```
1 <p>This is some
2 <app>
3   <lem>
4     <anchor xml:id="bnote" />dummy</lem>
5     <note type="fontium"
6       target="#range(right(bnote),left(enote))">Text of the
7       note.</note>
8     <rdg>pseudo</rdg>
9   </app>text to demonstrate how to insert a note in the
10  argument of the
11  <emph>lem</emph>command.
12  <anchor xml:id="#enote" /></p>
```

As can be seen from the TEI xml output above, the span of text the note is about has been carefully delimited by two anchors (ll. 4 and 12), the first of which falls within `<lem>` (l. 4), but `ekdosis` has taken care of moving the note itself out of this element (ll. 5–7). Otherwise, the TEI output would not be valid.

## 8 Poetry

### 8.1 The Standard *verse* Environment

In order to typeset verse texts or poems, L<sup>A</sup>T<sub>E</sub>X provides the standard *verse* environment. Within this environment, `\` is normally used to end lines, with the exception of the last line. As a result, stanzas are separated from one another by a blank line.

`ekdosis` provides `ekdverse` which is recommended for use in place of the standard *verse* environment. By default, `ekdverse` produces the same result as *verse*. However, `ekdosis` deviates a little from the standard usage for it needs all lines of poetry to be ended by `\` as a distinct marker. In the following listing, stanzas are visually separated from one another by an additional vertical space of 2 ex (l. 5). Between stanzas, `%` is used to prevent T<sub>E</sub>X from introducing a blank line. But a blank line—or even no blank line—would produce the exact same result:—

```
1 \begin{ekdverse}
2   It is an ancient Mariner,\
3   And he stoppeth one of three.\
4   ‘By thy long grey beard and glittering eye,\
5   Now wherefore stopp'st thou me?\
6   %
7   The Bridegroom's doors are opened wide,\
8   And I am next of kin;\
9   The guests are met, the feast is set:\
```

```

10 May'st hear the merry din.'\\
11 \end{ekdverse}

```

PDF output:—

```

1      It is an ancient Mariner,
2      And he stoppeth one of three.
3      ‘By thy long grey beard and glittering eye,
4      Now wherefore stopp’st thou me?

5      The Bridegroom’s doors are opened wide,
6      And I am next of kin;
7      The guests are met, the feast is set:
8      May’st hear the merry din.’

```

TEI xml output:—

```


<lg>
  <l>It is an ancient Mariner,</l>
  <l>And he stoppeth one of three.</l>
  <l>‘By thy long grey beard and glittering eye,</l>
  <l>Now wherefore stopp'st thou me?</l>
  <l>The Bridegroom's doors are opened wide,</l>
  <l>And I am next of kin;</l>
  <l>The guests are met, the feast is set:</l>
  <l>May'st hear the merry din.'</l>
</lg>


```

One would have expected here the `<lg>` element to be used as delimiter to encode the stanzaic verse forms. But as can be seen, only the outermost level of line group has been converted into TEI xml, let alone the vertical spacing between stanzas which has been ignored. This is because it is about as much as the standard `verse` environment provides.

## 8.2 The verse Package

*New feature v1.2* ekdosis can use the facilities offered by the excellent `verse` package<sup>50</sup> to which it adds a specific environment for the encoding of line groups such as stanzas. Furthermore, as the `verse` package provides its own numbering mechanism, the lines can be numbered independently of prose text.<sup>51</sup>

 For what is described in this section to operate, ekdosis must be loaded with the global option `poetry=verse` as explained above on page 7. Simply loading the `verse` package by means of `\usepackage` will have no effect.

 The foregoing does not apply if the `memoir` class be used.<sup>52</sup> In this case, ekdosis automatically uses the code provided by this class without the need to set the global option `poetry=verse`.

<sup>50</sup> Peter R. Wilson and Will Robertson, *The Verse package* (version 2.4b) [Aids for typesetting simple verse] (May 10, 2014), <http://www.ctan.org/pkg/verse>.

<sup>51</sup> See on page 52 for details.

<sup>52</sup> Lars Madsen and Peter R. Wilson, *The Memoir package* (version 3.7o) [Typeset fiction, non-fiction and mathematical books] (Mar. 23, 2021), <http://www.ctan.org/pkg/memoir>.

**Compatible Verse Commands** The reader is invited to refer to the documentation of the `verse` package for detailed information. Within the `ekdverse` environment, `\` *must be used* at the end of each line, as follows:—

- (a) `\` is the standard command to be used at the end of each line.
- (b) `\!` must be used at the end of stanzas or line groups instead of `\`.
- (c) `\*` does the same as `\` except that it prohibits a page break after the line.
- (d) `\>` is for line breaks within a verse line.

*New feature v1.4* (e) `\+` does the same as `\>` but without indenting the subsequent line which further complies to any already defined indent pattern.

All of these commands can take a dimension as optional argument, like so: `\[30pt]`, `\![30pt]`, `\*[30pt]`, `\>[30pt]` or `\+[30pt]`. If `\`, `\!`, `\*` or `\+` be used, a vertical space of the dimension specified is added between lines, whereas `\>[...]` adds an horizontal space after the line break.

`\vin` `\vin` indents a verse line by a length which is by default 1.5 em. This length is stored as `\vgap` and can be changed by `\setlength` or `\addtolength`.

`ekdverse (env.)` **The `ekdverse` Environment** `\begin{ekdverse}[\langle options \rangle] ... \end{ekdverse}`

This environment is used to hold verse lines as described above and may receive an optional argument in which the following “name=value” arguments are accepted:—

`width` `width=\langle length \rangle` Default: `\linewidth`


If `width` be supplied, it is taken as a length in relation to which the entire contents of the environment are to be horizontally centered. If given, this dimension may correspond to an average line or to the longest line of the line group. To this end, the standard L<sup>A</sup>T<sub>E</sub>X command `\settowidth` can be used, like so:—

```
\settowidth{\versewidth}{This is the average line,}
\begin{ekdverse}[width=\versewidth]
...
\end{ekdverse}
```

REM. `\versewidth` is provided by the `verse` package as a convenience and can be used by `ekdosis`.

`type` `type=\langle type \rangle` Default: not set

This named argument is used in the TEI `xml` output to name the type of unit encoded within the `<lg>` element, viz. “sonnet”, “quatrain”, “couplet” and the like.

 Unlike the TEI `xml` element `<lg>`, `ekdverse` may not nest hierarchically. Within this environment, `ekdstanza` must be used instead to encode stanzas as described below on page 47.

As an example, the first five lines of Homer’s *Odyssey* could be encoded like so.<sup>53</sup>—

```
\begin{alignment}[tcols=2,
                    lcols=2,
                    texts=homer[xml:lang="grc"];murray[xml:lang="en"],
                    apparatus=homer]
\begin{homer}
\begin{ekdverse}
Ἄνδρα μοι ἔννεπε, Μοῦσα, πολύτροπον, ὃς μάλα πολλά \
πλάγχθη, ἐπεὶ Τροίης ἱερὸν πτολίεθρον ἔπερσεν \
πολλῶν δ' ἀνθρώπων ἴδεν ἄστεα καὶ
\app{\lem{νόον}
\rdg[resp=Zen]{νόμον}
```

53. Homer, *The Odyssey*, ed. A. T. Murray, 2 vols. (Cambridge, MA. – London: Harvard University Press – William Heinemann, 1919).



```

\textnote{Cf. Schol.}} ἔγνω, \\
πολλὰ δ' ὁ γ' ἐν πόντῳ πάθεν ἄλγεα ὄν κατὰ θυμόν, \\
ἀρνύμενος ἦν τε ψυχὴν καὶ νόστον ἐταίρων. \\
\end{ekdverse}
\end{homer}
\begin{murray}
Tell me, O Muse, of the man of many devices, who wandered full
many ways after he had sacked the sacred citadel of Troy. Many
were the men whose cities he saw and whose mind he learned, aye,
and many the woes he suffered in his heart upon the sea, seeking
to win his own life and the return of his
comrades.
\end{murray}
\end{alignment}

```

PDF output:—

<p>Ἄνδρα μοι ἔννεπε, Μοῦσα, πολύτροπον, ὃς μάλα πολλὰ 1  πλάγχθη, ἐπεὶ Τροίης ἱερὸν πτολίεθρον ἔπερσεν· 2  πολλῶν δ' ἀνθρώπων ἴδεν ἄστεα καὶ νόον ἔγνω, 3  πολλὰ δ' ὁ γ' ἐν πόντῳ πάθεν ἄλγεα ὄν κατὰ θυμόν, 4  ἀρνύμενος ἦν τε ψυχὴν καὶ νόστον ἐταίρων. 5</p> <hr/> <p>3 νόον] νόμον Zen. Cf. Schol.</p>	<p>Tell me, O Muse, of the man of 1  many devices, who wandered full 2  many ways after he had sacked 3  the sacred citadel of Troy. Many 4  were the men whose cities he saw 5  and whose mind he learned, aye,  and many the woes he suffered in  his heart upon the sea, seeking to  win his own life and the return of  his comrades.</p>
--	---

TEI xml output:—

```

<div xml:id="div-homer_1" xml:lang="grc">
  <lg>
    <l>Ἄνδρα μοι ἔννεπε, Μοῦσα, πολύτροπον, ὃς μάλα πολλὰ</l>
    <l>πλάγχθη, ἐπεὶ Τροίης ἱερὸν πτολίεθρον ἔπερσεν.</l>
    <l>πολλῶν δ' ἀνθρώπων ἴδεν ἄστεα καὶ
    <app>
      <lem>νόον</lem>
      <rdg resp="#Zen">νόμον</rdg>
      <note>Cf. Schol.</note>
    </app>ἔγνω,</l>
    <l>πολλὰ δ' ὁ γ' ἐν πόντῳ πάθεν ἄλγεα ὄν κατὰ θυμόν,</l>
    <l>ἀρνύμενος ἦν τε ψυχὴν καὶ νόστον ἐταίρων.</l>
  </lg>
</div>
<div xml:id="div-murray_1" xml:lang="en">
  <p>Tell me, O Muse, of the man of many devices, who
  wandered full many ways after he had sacked the sacred
  citadel of Troy. Many were the men whose cities he saw and
  whose mind he learned, aye, and many the woes he suffered
  in his heart upon the sea, seeking to win his own life and
  the return of his comrades.</p>
</div>

```

**Stanzas** As can be seen above, the L<sup>A</sup>T<sub>E</sub>X `ekdverse` environment is translated into the TEI `xml` `<lg>` element. The `type` attribute may then be used to name the type of unit encoded by this element.

```
ekdstanza (env.) \begin{ekdstanza}[\langle options \rangle] ... \end{ekdstanza}
```

This environment is used within `ekdverse` to encode succeeding stanzaic forms. Within `ekdstanza`, the last line is ended by `\!` or `\!` depending on whether an additional vertical space is required between stanzas. This environment may receive an optional argument in which the following “name=value” argument is accepted:—

```
type type=\langle type \rangle
```

Default: not set

As in the case of `ekdverse`, this named argument is used in the TEI `xml` output to name the type of unit encoded within the `<lg>` element, viz. “quatrain”, “couplet” and the like.

**Indentation Patterns** `ekdosis` can use the `patverse` environment and its associated command `\indentpattern` that are provided by the `verse` package. As described in the documentation of this package,<sup>54</sup> the indentation pattern consists of an array of digits,  $d_1$  to  $d_n$ , where the  $n^{\text{th}}$  line is indented by  $d_n$  times the amount of `\vgap` described above on page 45.

The overall structure of lines grouped into stanzas may look as follows:—

```
1 \begin{ekdverse}[type={overall type}]
2   \indentpattern{digits}
3   \begin{patverse}
4     \begin{ekdstanza}[type={stanza 1 type}]
5       line 1 \\
6       line 2 \\
7       [...]
8       final line \\!
9     \end{ekdstanza}
10  \end{patverse}
11 \end{ekdverse}
```

Of course, if no indentation pattern be required or be only required occasionally, `patverse` (ll. 3 and 10) and `\indentpattern` (l. 2) are of no use:—

```
1 \begin{ekdverse}[type={overall type}]
2   \begin{ekdstanza}[type={stanza 1 type}]
3     line 1 \\
4     line 2 \\
5     \vin indented line 3 \\
6     [...]
7     final line \\!
8   \end{ekdstanza}
9 \end{ekdverse}
```

A detailed example follows. It is taken from Raymond MacDonald Alden’s edition of Shakespeare’s *Sonnets* from the Quarto of 1609 with variorum readings.<sup>55</sup> Compared to MacDonald’s edition, an effort has been made to use the typography and punctuation of the original edition which can be consulted online at the British Library’s website.<sup>56</sup> However, this typographical refinement has been retained for the sonnet only. For the sake of clarity,

54. Wilson and Robertson (cf. n. 50), 6.

55. Shakespeare, *The Sonnets* [From the Quarto of 1609 with Variorum Readings and Commentary], ed. Raymond MacDonald Alden (Boston & New York: The Riverside Press Cambridge, 1916), Sonnet 1, p. 15.

56. Shake-speares *Sonnets*. Neuer before Imprinted. (A Louers Complaint. By William Shake-speare.). <https://www.bl.uk/collection-items/first-edition-of-shakespeares-sonnets-1609>. Call number C.21.c.44, fol. B.

the line numbers and the apparatus criticus use modern typography. The references to line numbers in the apparatus criticus have been made consistent with MacDonald's edition, as have the entries in the apparatus criticus, namely the bare line number followed by a dot, then the lemma text in bold face, then the variant in italic shape:—

Listing 8: Poetry: Shakespeare's Sonnet 1

```

1  \junicode % Use the Junicode font with 'hist' feature enabled for
2      % long-s
3  % MacDonald's style for numbers and entries in the apparatus
4  % criticus:
5  \SetHooks{
6      refnumstyle=\normalfont,
7      postrefnum=.,
8      lemmastyle=\bfseries,
9      readingstyle=\itshape
10 }
11 % The lines are to be centered horizontally:
12 \settowidth{\versewidth}{Feed'st thy lights flame with selfe
13 substantiall fewell,}
14 % Format of the outermost <div> element:
15 \NewDocumentEnvironment{ekdcenter}{-}{\par\centering}{\nobreak\par}
16 \FormatDiv{1}{\begin{ekdcenter}}{\end{ekdcenter}}
17 \begin{ekdosis}
18     \ekddiv{type=sonnets, n=1, head=1}
19     \begin{ekdverse}[type=sonnet, width=\versewidth]
20         \indentpattern{00000000000011}
21         \begin{patverse}
22             \begin{ekdstanza}[type=quatrain]
23                 \ekdlettrine{F}{r}om fairest creatures we desire
24                 increase,\
25                 That thereby beauties \emph{Rose}
26                 \app{
27                     \lem{might}
28                     \rdg[source={Gildon1710, Sewell1725, Ewing1771}]{may}
29                 } neuer die,\
30                 But as the riper should by time
31                 \app{
32                     \lem{decease}
33                     \rdg[source={Hudson1856}]{decease}
34                 },\
35                 His tender heire might beare his memory:\
36             \end{ekdstanza}
37             \begin{ekdstanza}[type=quatrain]
38                 But thou contracted to thine owne bright eyes,\
39                 Feed'st thy
40                 \app{
41                     \lem{lights}
42                     \rdg[source={Butler1899, Walsh1908}]{life's}
43                 } flame with
44                 \app{
45                     \lem{selfe substantiall}
46                     \rdg[source={Gildon1714,
47                         alt={\textnormal{Hyphened by}}]{selfe-substantiall}
48                     \note{etc.}} fewell,\
49                 Making a famine where abundance lies,\

```

```

50     Thy selfe thy foe,to thy sweet selfe too cruell:\\
51 \end{ekdstanza}
52 \begin{ekdstanza}[type=quatrain]
53     Thou that art now the worlds fresh ornament,\\
54     And
55     \app{
56         \lem{only}
57         \rdg[resp=God, type=conjecture]{early}
58         \note{conj.}
59     } herauld to the gaudy spring,\\
60     Within thine owne bud buriest thy content,\\
61     And tender
62     \app{
63         \lem{chorle}
64         \rdg[source=Gildon1710]{churl}
65         \note{etc.}
66     } makst wast in niggarding:\\
67 \end{ekdstanza}
68 \begin{ekdstanza}[type=couplet]
69     Pitty the world,or else this glutton be,\\
70     To eate the worlds due,\app{
71         \lem{by the}
72         \rdg[resp=Stee, type=conjecture]{be thy}
73         \note{conj.}
74         \rdg[resp=God, type=conjecture]{by thy}
75         \note{conj.}
76     } graue
77     \app{
78         \lem{and}
79         \rdg[resp=God, type=conjecture]{as}
80         \note{conj.}
81     } thee.\\!
82 \end{ekdstanza}
83 \end{patverse}
84 \end{ekdverse}
85 \end{ekdosis}

```

REM. 1 Gildon1710, Gildon1714, Sewell1725, Ewing1771, Hudson1856, Butler1899 and Walsh1908 have been declared as sources.<sup>57</sup> God and Stee, Godwin and Steevens, have been declared as scholars.<sup>58</sup>

REM. 2 `\ekdletterine` (l. 23) is a specific command for the letterine package does not work in list environments. `\TeXtoTEIPat` has been used as described below on page 73 to instruct `ekdosis` to convert this command into an acceptable TEI equivalent. The definition of `\ekdletterine` follows:—

```

% Preamble:
\usepackage{adjustbox}
% This basic command actually requires an adjustment of the vertical
% space at the end of the current line (eg. \[-1.875ex]) and \vin at
% the beginning of the next line:---
\NewDocumentCommand{\ekdletterine}{mm}{%
\adjustbox{valign=t,raise=-0.75ex}{\Huge #1}\textsc{#2}%
}
\TeXtoTEIPat{\ekdletterine {#1}{#2}}{<hi rend="smallcaps">#1#2</hi>}

```

57. See above on page 10 and below sect. 13.7 on page 76.

58. See above on page 11.

PDF output:—

	1	
<b>F</b> ROM fairest creatures we desire increase,	1	
That thereby beauties <i>Rose</i> might neuer die,	2	
But as the riper should by time decease,	3	
His tender heire might beare his memory:	4	
But thou contracted to thine owne bright eyes,	5	
Feed'st thy lights flame with selfe substantiall fewell,	6	
Making a famine where abundance lies,	7	
Thy selfe thy foe, to thy sweet selfe too cruell:	8	
Thou that art now the worlds fresh ornament,	9	
And only herauld to the gaudy spring,	10	
Within thine owne bud burieft thy content,	11	
And tender chorde makst wast in niggarding:	12	
Pitty the world, or else this glutton be,	13	
To eate the worlds due, by the graue and thee.	14	

---

2. might] *may* G S E 3. decease] *decrease* Hu<sup>2</sup> 6. lights] *life's* But Wa selfe substantiall] Hyphened by G<sup>2</sup> etc. 10. only] *early* Godwin conj. 12. chorle] *churl* G etc. 14. by the] *be thy* Stee conj. *by thy* Godwin conj. and] *as* Godwin conj.

TEI xml output:—


```
<div type="sonnets" n="1">
  <head>1</head>
  <lg type="sonnet">
    <lg type="quatrain">
      <l>
        <hi rend="smallcaps">Fr</hi>om fairest creatures we
        desire increase,</l>
      <l>That thereby beauties
        <emph>Rose</emph>
        <app>
          <lem>might</lem>
          <rdg source="#Gildon1710 #Sewell1725 #Ewing1771">
            may</rdg>
        </app>neuer die,</l>
      <l>But as the riper should by time
        <app>
          <lem>decease</lem>
          <rdg source="#Hudson1856">decrease</rdg>
        </app>,</l>
      <l>His tender heire might beare his memory:</l>
    </lg>
    <lg type="quatrain">
      <l>But thou contracted to thine owne bright eyes,</l>
      <l>Feed'st thy
        <app>
          <lem>lights</lem>
          <rdg source="#Butler1899 #Walsh1908">
            life's</rdg>
        </app>
      </l>
    </lg>
  </lg>
</div>
```

```

</app>flame with
<app>
  <lem>selfe substantiall</lem>
  <rdg source="#Gildon1714">
    selfe-substantiall</rdg>
  <note>etc.</note>
</app>fewell,</l>
<l>Making a famine where aboundance lies,</l>
<l>Thy selfe thy foe,to thy sweet selfe too
cruell:</l>
</lg>
<lg type="quatrain">
  <l>Thou that art now the worlds fresh ornament,</l>
  <l>And
  <app>
    <lem>only</lem>
    <rdg resp="#God" type="conjecture">early</rdg>
    <note>conj.</note>
  </app>herauld to the gaudy spring,</l>
  <l>Within thine owne bud buriest thy content,</l>
  <l>And tender
  <app>
    <lem>chorle</lem>
    <rdg source="#Gildon1710">churl</rdg>
    <note>etc.</note>
  </app>makst wast in niggarding:</l>
</lg>
<lg type="couplet">
  <l>Pitty the world,or else this glutton be,</l>
  <l>To eate the worlds due,
  <app>
    <lem>by the</lem>
    <rdg resp="#Stee" type="conjecture">be thy</rdg>
    <note>conj.</note>
    <rdg resp="#God" type="conjecture">by thy</rdg>
    <note>conj.</note>
  </app>graue
  <app>
    <lem>and</lem>
    <rdg resp="#God" type="conjecture">as</rdg>
    <note>conj.</note>
  </app>thee.</l>
</lg>
</lg>
</div>

```

## 9 Lination Settings

 ekdosis uses lineno internally for line numbering.<sup>59</sup> But it must be noted that ekdosis strictly prohibits the “pagewise” mode of operation that is provided by lineno. As a

<sup>59</sup> Uwe Lück and Stephan Böttcher, *The Lineno package* (version 4.41) [Line numbers on paragraphs] (Nov. 2, 2005), <http://www.ctan.org/pkg/lineno>.

result of this hinderance, all “margin switching” functions of `lineno` are disabled within the environments that are specific to `ekdosis`, viz. `ekdosis` and `alignment`.

That said, `ekdosis` provides equivalents of its own to handle the line numbers the same way as `lineno`’s “pagewise” mode of operation does.

`\SetLineation` `\SetLineation{csv list of options}` may be used in the preamble or at any point of the document to set lineation preferences. Its argument processes the `key=value` options that follow:—

### General Options

`lineation` `lineation=page|document|none` Default: document  
`lineation=document` has the lines numbered continuously throughout the document while `lineation=page` instructs `ekdosis` that the numbering should start afresh at the top of each page. `none` does the same as `page` but prevents the numbers from being printed in the margins while keeping them in use in the apparatus criticus.

`modulo` `modulo` Default: not set  
`modulo` does not accept any value. When this option is set, every fifth line is numbered.

`modulonum` `modulonum=n` (where *n* is an integer) Default: not set  
`modulonum` allows to modify the interval between the numbers that are printed. `modulo` must be set for this option to have effect. As examples, `modulo`, `modulonum=3` has every third line numbered and `modulonum=1` disables `modulo` numbering.

`margin` `margin=right|left|inner|outer` Default: left  
`margin` sets the margin in which the line numbers are to be printed.

`numbers` `numbers=elided|full` Default: elided  
This option only has effect on the numbers that are printed in the apparatus criticus. `numbers=elided` applies on spans of numbers and elides the last number of a range to the fewest number of figures possible—viz. 35–7, 129–31 *&c.*—without eliding digits in the group 10 to 19 in each hundred—viz. 17–19, 115–18 *&c.* `numbers=full` leaves the numbers untouched.

`\innerlinenumbers` `\innerlinenumbers` and `\outerlinenumbers` are equivalent to `\SetLineation{numbers=outer}` and `\SetLineation{numbers=inner}` respectively. Both commands are complementary to `\rightlinenumbers` and `\leftlinenumbers` already provided by the `lineno` package.

### Options Specific to the `poetry=verse` Global Setting<sup>60</sup>

`vlineation` `vlineation=page|document` Default: document  
`vlineation` operates on verse texts in the same way as `lineation` on prose texts.

`vmodulo` `vmodulo=n` (where *n* is an integer) Initially: 1, Default: 5  
`vmodulo=n` has every *n*<sup>th</sup> lines of verse printed in the margin. If used without value, this option is equivalent to `vmodulo=5`. `vmodulo=0` prevents the numbers from being printed.

`vnumbrokenlines` `vnumbrokenlines=true|false` Initially: false

*New feature v1.4* This named argument does not need a value as it defaults to `true` if used. `vnumbrokenlines` has both parts of lines broken by `\>` or `\>+61` numbered with the same number for disambiguation purposes. By default, only the second part of broken lines is numbered.

`vmargin` `vmargin=right|left` Default: right  
`vmargin` sets the margin in which the verse line numbers are to be printed.

`continuousvnum` Default: not set  
The `poetry=verse` global option has the succeeding lines of verse numbered independently of prose text. `continuousvnum` has all lines numbered continuously, irrespective of whether they are lines of prose or poetry text.

60. See above [sect. 8.2 on page 44](#).

61. See above [\(d\)](#) to [\(e\)](#) on page 45 for more information.

**Useful Lineation Commands** As implied above, pretty much all commands that are provided by the “running” mode of operation of the `lineno` package will work with `ekdosis`, notably the following:—

`\modulolinenumbers` `\modulolinenumbers[⟨n⟩]` can be used to enable or modify modulo line numbering as described above.

`\resetlinenumber` `\resetlinenumber[⟨n⟩]` resets the line number to one or to  $n$  if specified.

`\linenumberfont` `\renewcommand{\linenumberfont}{⟨commands⟩}` can be used to set the font used for the line numbers that are printed in the margins. By default, the definition is `\normalfont\footnotesize`.

`\linenumbersep` `\linenumbersep` is the distance between the numbers and the margin. By default, this distance is set to 10 pt. It can be redefined like so: `\setlength\linenumbersep{⟨length⟩}`.

**Poetry Lineation Commands** In addition to these commands, if `ekdosis` be loaded with the global option `poetry=verse` as described above in [sect. 8.2 on page 44](#), the commands dedicated to line numbering that are provided by the `verse` package can be used.

`\modulolinenumbers` `\modulolinenumbers[⟨n⟩]` can be used to enable or modify verse modulo line numbering as described above.

*New feature v1.4*

`\resetvlinenumber` `\resetvlinenumber[⟨n⟩]` has for lines of verse the same effect as `\resetlinenumber` for lines of prose text.

`\verselinenumfont` `\verselinenumfont{⟨commands⟩}` can be used to set the font used for lines of verse. By default, the definition is `\normalfont\footnotesize`.


`\vrightskip` The `verse` package prints the numbers at the distance `\vrightskip` into the right margin. This distance can be redefined by means of `\setlength` like so: `\setlength\vrightskip{⟨length⟩}`.

`\linelabel` **Labels** In prose as well as in poetry texts, `\linelabel{⟨label⟩}` sets a line label that can be referred to with `\ref{⟨label⟩}`.

As an example, what follows has every fifth line number printed in the inner margins. Additionally, the numbering shall start afresh at the top of each page:—

```
\SetLineation{
  lineation=page,
  modulo,
  margin=inner
}
```


## 10 Languages

 `ekdosis` is fully compatible with `babel`. “Fully compatible” means that all features provided by `babel`, including language switching commands, are supported by `ekdosis`. `ekdosis` is also compatible with `polyglossia` with one notable exception: `luabidi`, which `polyglossia` loads for languages written from right to left, is not supported by `ekdosis`, and most probably never will be. That said, as far as the author could see, single-layer apparatus, as described in [sect. 6.2 on page 36](#), can be typeset within the Arabic environment that is provided by `polyglossia`. Unfortunately, the same cannot be said for multiple-layer apparatus.

Whether `babel` or `polyglossia` is used, `ekdosis` automatically applies the current language to the entries of the apparatus criticus, including the fonts that may have been associated to the languages in the preamble. In this respect, as `polyglossia` can use the same language




switching commands as `babel`,<sup>62</sup> the general advice given above in [sect. 5.1.1 on page 32](#) applies in all cases. As regards setting languages in the TEI `xml` output file, the reader is invited to refer to [point \(c\) on page 30](#), and [sect. 13 on page 67](#) including the example provided on [page 73](#).

 In some cases, it may be needed to apply in the apparatus criticus a language different from the one that is selected in the edition text. To this end, `ekdosis` provides a set of facilities which are described on [pages 36–37](#) for single-layer apparatus and on [page 38](#) for multiple-layer apparatus.

## 10.1 Languages Written From Right to Left

As said above, `polyglossia` is not supported by `ekdosis` for languages that are written and read from right to left, like Arabic, Hebrew or Syriac. However, as `babel` is supported and can be loaded concurrently with `polyglossia`, an easy way is to use `babel` to print such languages.

 The reader is invited to refer to and become acquainted with the relevant parts of the documentation of the `babel` package.<sup>63</sup>

**babel Only** In the following example, `babel` is used exclusively to set three different languages: Arabic, ancient Greek and English:—

Listing 9: Multilingual editions with `babel` only

```

1  \usepackage{fontspec}
2
3  \usepackage[greek,ancient,english]{babel}
4  \babelprovide[onchar=fonts]{arabic}
5
6  \babelefont{rm}{Old Standard}
7  \babelefont[greek]{rm}[RawFeature={+ss05;+ss06}]{Old Standard}
8  \babelefont[*arabic]{rm}{Amiri}
9
10 \babeletags{ancientgreek = greek}
11 \newcommand{\sg}[1]{\textancientgreek{#1}}
12
13 \newcommand{\RL}[1]{\bgroup\textdir TRT#1\egroup}
14 \newenvironment{Arabic}{\par\pardir TRT\textdir TRT}{\par}

```

REM. 1 As can be seen, `fontspec` has been loaded before `babel`. To the author’s knowledge, this gives better results when `\babelefont` is used.

REM. 2 Line 3 loads `babel` and instructs it to use English as the default language and ancient Greek as a second optional language. The built-in  `bidi`  mechanism provided by `babel` is not enabled. As a result, specific language switching commands for Arabic must be defined just as it must be for every other language.

REM. 3 Line 4 does not load any Arabic, but instructs `babel` that it should use the Arabic font that is set below with `\babelefont` whenever an Arabic letter is encountered.

REM. 4 Lines 6–8 select the fonts: `Old Standard` is the default font to be used for Roman shape (l. 6); the same font is used for Greek, with some additional Open Type features enabled; finally, the `Amiri` font is used for Arabic.

REM. 5 Lines 10–11 define so-called “tags” for easier access to ancient Greek through `\begin{ancientgreek} ... \end{ancientgreek}` for running paragraphs and `\textancientgreek{<text>}` for short insertions of Greek in English text. `\sg{<text>}` is just a shorthand for this latter command.

62. See François Charette and Arthur Reutenauer, *The Polyglossia package* (version 1.49) [An alternative to `babel` for XeLaTeX and LuaLaTeX] (Apr. 8, 2020), <http://www.ctan.org/pkg/polyglossia>, 3.2 p. 14.

63. Javier Bezos López and Johannes L. Braams, *The Babel package* (version 3.47) [Multilingual support for Plain TeX or LaTeX] (July 13, 2020), <http://www.ctan.org/pkg/babel>.

REM. 6 Finally, lines 13–14 define simple language switching commands for Arabic. As can be seen, no commands other than the LuaTeX primitives `\pardir` and `\textdir` have been used for `babel` already takes care of selecting the Arabic font. `\RL` is for short insertions of Arabic words in English paragraphs while `\begin{Arabic} ... \end{Arabic}` is for running paragraphs of Arabic text.

`\setRL` **Changing the Writing Direction** `\setRL` and `\setLR` are two argument-less commands provided by `ekdosis` that can be used to change the writing direction of running paragraphs. The former sets the direction from right to left and the latter from left to right. If `babel` be set as above, `\setRL ... \setLR` can be used in place of `\begin{Arabic} ... \end{Arabic}`.

**polyglossia Associated With `\babelprovide`** What follows illustrates how `babel` can be used conjointly with `polyglossia` for the same three languages as above without having to load `luabidi`:—

Listing 10: Multilingual editions with `babel` and `polyglossia`

```

1  \usepackage{fontspec}
2
3  \usepackage{babel}
4  \babelprovide[onchar=fonts]{arabic}
5
6  \setmainfont{Old Standard}
7  \newfontfamily\greekfont{Old Standard}[RawFeature={+ss05;+ss06}]
8  \babelfont[*arabic]{rm}{Amiri}
9
10 \usepackage{polyglossia}
11 \setdefaultlanguage{english}
12 \setotherlanguage[variant=ancient]{greek}
13
14 \newcommand{\textarabic}[1]{\bgroup\textdir TRT#1\egroup}
15 \newenvironment{Arabic}{\par\pardir TRT\textdir TRT}{\par}

```

REM. 1 Line 3 just loads `babel` with no default language.

REM. 2 Lines 4 and 8 are used to have the Arabic font automatically selected as above.

REM. 3 Lines 14–15 define the exact language switching commands that would have been defined if `polyglossia` and `luabidi` had been used for Arabic.

As one can see, the important points about languages written from right to left are to use `babel` only to select the Arabic fonts, avoid using the bidirectional mechanism it provides and define commands and environments that use only LuaTeX primitives to set the writing direction. Then, an Arabic edition text—to continue with this example—can be entered as plainly as follows:—

```

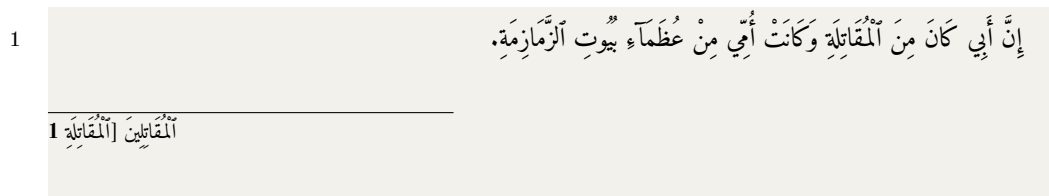
\begin{ekdosis}
  \begin{Arabic}
    إِنَّ أَيَّ كَانَ مِنَ
    \lem{المقاتلة}
    \rdg{المقاتلين}
    وَكَانَتْ أُمِّي مِنْ عُظَمَاءِ بَيْتِ الزَّمَاةِ.
  \end{Arabic}
\end{ekdosis}

```

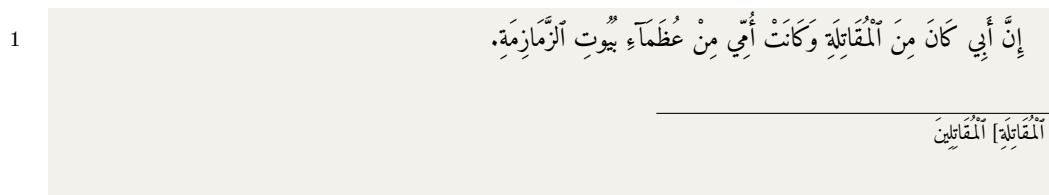
It should be reminded that the writing direction of the apparatus criticus itself is independent of that of the edition text and must be set either with `\Set(LTR|RTL)app`

or with the `direction` optional argument of `\SetApparatus` for single-layer apparatus criticus, or by means of `\DeclareApparatus` for multiple-layer apparatus criticus.<sup>64</sup>

The PDF output with left-to-right apparatus criticus follows:—



And here follows the PDF output with right-to-left apparatus criticus:—



## 10.2 Using arablutax

arablutax is a Lua $\text{\LaTeX}$  package that provides commands and environments which return Arabic writing from an ASCII transliteration (either Arab $\text{\TeX}$  or Buckwalter scheme).<sup>65</sup> It is particularly well-suited for complex documents such as critical editions where a lot of commands intertwine with Arabic writing. arablutax can output Unicode Arabic in the same modes as arab $\text{\TeX}$ <sup>66</sup> or in different accepted standards of romanization. It is also able to produce a duplicate of the original `.tex` source file in which all arab $\text{\TeX}$  or buckwalter strings are replaced with Unicode equivalents, either in Arabic script or in any accepted standard of transliteration.<sup>67</sup>

arablutax is fully supported by ekdosis. The following example illustrates how arablutax and ekdosis interact with each other to produce distinct TEI `xml` outputs from a single `.tex` source file:—

Listing 11: ekdosis and arablutax

```

1 % Preamble:
2 % load ekdosis and ask for TEI xml output:
3 \usepackage[telexport]{ekdosis}
4 % load arablutax and request a LaTeX output with Unicode Arabic:
5 \usepackage[export,fullvoc]{arablutax}
6
7 % document:
8 \begin{arabexport} % export arabtex strings to Unicode Arabic
9   \begin{ekdosis}
10    \begin{arab}
11      'inna 'abI kAna mina
12    \app{

```

64. See above [sect. 6.2 on page 36](#) (single-layer apparatus criticus) and [sect. 6.3 on page 38](#) (multiple-layer apparatus criticus).

65. Robert Alessi, *The Arablutax package* (version 1.20) [Arab $\text{\TeX}$  for Lua $\text{\LaTeX}$ ] (Mar. 23, 2020), <http://ctan.org/pkg/arablutax>.

66. Klaus Lagally, *The Arabtex package* (version 4.00) [Macros and fonts for typesetting Arabic] (Mar. 11, 2004), [http://baobab.informatik.uni-stuttgart.de/ifi/bs/research/arab\\_e.html](http://baobab.informatik.uni-stuttgart.de/ifi/bs/research/arab_e.html).

67. Alessi, *The Arablutax package* (cf. n. 65), “Exporting Unicode Arabic to an External File.”

```

13     \lem{'l-muqAtilaTi}
14     \rdg{'l-muqAtilIna}
15   }
16   wa-kAnat 'ummI min `u.zamA'i buyUti 'l-zamAzimaTi.
17   \end{arab}
18   \end{ekdosis}
19 \end{arabexport}

```

The PDF output with left-to-right apparatus criticus is of course the same as above:—

1 إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ وَكَانَتْ أُمِّي مِنْ عُظْمَاءِ بِيوتِ الزَّمَاذِمَةِ.

---

المقاتلين | المقاتلة | 1

However, assuming that the source file is called `source.tex`, `ekdosis` produces as instructed from this file an additional `source-tei.xml` as follows:—

```

<p xml:lang="ar-Latn" type="transliterated"
subtype="arabtex">'inna 'abI kAna mina
<app>
  <lem>'l-muqAtilaTi</lem>
  <rdg>'l-muqAtilIna</rdg>
</app>wa-kAnat 'ummI min `u.zamA'i buyUti
'l-zamAzimaTi.</p>

```

At the same time, `arabluatex` is instructed to produce on its own from the same `source.tex` an additional `source_out.tex` in which all `arabtex` strings found within `\begin{arabexport} ... \end{arabexport}` (see [listing 11 on the previous page](#), ll. 9–19) are replaced with full-vocalized Arabic Unicode script. Finally, compiling this latter file produces the following `sample-arabic_out-tei.xml` an extract of which follows:—

```

<p xml:lang="arb">إِنَّ أَبِي كَانَ مِنَ الْمُقَاتِلَةِ وَكَانَتْ أُمِّي مِنْ عُظْمَاءِ بِيوتِ الزَّمَاذِمَةِ.
<app>
  <lem>المقاتلة</lem>
  <rdg>المقاتلين</rdg>
</app>وَكَانَتْ أُمِّي مِنْ عُظْمَاءِ بِيوتِ الزَّمَاذِمَةِ.
</p>

```

The reader will find the full `arabic-sample.tex` source file with instructions in [sect. 18 on page 97](#), and is invited to refer to the documentation of the `arabluatex` package for more information on the way to use its Arabic environments and built-in functions dedicated to export `arabtex` ASCII strings to Unicode.<sup>68</sup>

## 11 Divisions of the Body

The features that are described in this section call for one general remark. `ekdosis` is designed to figure out where any `LATEX` command that is converted to a TEI opening

<sup>68</sup>. Alessi, *The Arabluatex package* (cf. n. 65).

element allowed to nest recursively, such as `<div>`, `<lg>` and the like, is to be closed, even though there is no explicit indication of the point where the closure occurs. Thoroughly scanning L<sup>A</sup>T<sub>E</sub>X source files with Lua functions which involve complex string matching and recursions was required, as L<sup>A</sup>T<sub>E</sub>X ‘open’ commands such as `\chapter` or `\section` only act as milestones, contrary to TEI elements.

⚠ It must be noted that the two styles described hereinafter are mutually exclusive. TEI `xml` forbids that both be combined within a single `<body>` element.<sup>69</sup> As a result, `ekdosis` will disregard whichever one is not selected.

## 11.1 L<sup>A</sup>T<sub>E</sub>X Standard Divisions

`ekdosis` can use the L<sup>A</sup>T<sub>E</sub>X standard textual divisions, such as `\book`, `\chapter`, `\section` and the like.

⚠ However, to have these divisions properly translated into TEI numbered `<div>` elements, the `divs` general option must be set to `latex` explicitly—viz. `divs=latex`—as described above on page 7.

⚠ As the `alignment` environment that is provided by `ekdosis` places all aligned texts within TEI `xml` un-numbered `<div>` elements and L<sup>A</sup>T<sub>E</sub>X textual divisions are converted into numbered `<divn>` elements, inserting such divisions in texts to be aligned will result in an invalid TEI `xml` output. Instead, un-numbered divisions through `\ekddiv` must be used as described below in [sect. 11.2 on the following page](#).

Once `divs` has been set to `latex`, `ekdosis` converts `\book`, `\part`, `\chapter`, `\section`, `\subsection` and `\subsubsection` into corresponding TEI ‘numbered’ `<divn>` elements, where  $1 \leq n \leq 6$ .

`\MkBodyDivs` **Adjusting the Levels of Textual Subdivisions** `\MkBodyDivs{<div1>}{<div2>}{<div3>}{<div4>}{<div5>}{<div6>}` takes six mandatory arguments. This command can be used in the preamble or at any point of the document to make the number of the first-level subdivision of the edition text, viz. `<div1>`, match to any L<sup>A</sup>T<sub>E</sub>X command other than `\book`. For example, if `\section` be the highest-level sectional command used, then `\MkBodyDivs{section}{subsection}{subsubsection}{}{}{}` will have `\section`, `\subsection` and `\subsubsection` converted into `<div1>`, `<div2>` and `<div3>` respectively.

**Inserting Variants in Headings** Variant readings can be inserted in headings. In this case, the optional argument of the L<sup>A</sup>T<sub>E</sub>X sectional command must naturally be used to prevent variants from going into headers, footers or the table of contents, like so:<sup>70</sup>—

```

1 % Preamble:
2 \usepackage[telexport=tidy, divs=latex]{ekdosis}
3 \MkBodyDivs{chapter}{section}{}{}{}{}
4
5 % Document:
6 \chapter[Ἰπποκράτους ἐπιδημιῶν βιβλίον δεῦτερον]{Ἰπποκράτους ἐπιδημιῶν
7   \app{
8     \lem[wit={I,R,H}]{βιβλίον δεῦτερον}
9     \rdg[wit=V]{λόγος β'}}}
10
```

69. See <https://tei-c.org/release/doc/tei-p5-doc/en/html/DS.html#DSDIV>.

70. On the use of `egomute` (l. 13), see above [REM. 2 on page 22](#).

```

11 \section[Τμήμα πρώτον]{
12   \app{
13     \lem[resp=egomute, type=emendation, nosep,
14       post=suppleui]{\supplied{Τμήμα πρώτον}}
15   }}
16 Άνθρακες θερινοὶ ἐν Κραννῶνι. [...]

```


TEI xml output:—

```

<div1 type="chapter">
  <head>Ἰπποκράτους ἐπιδημιῶν
  <app>
    <lem wit="#I #R #H">βιβλίον δεύτερον</lem>
    <rdg wit="#V">λόγος β'</rdg>
  </app></head>
  <div2 type="section">
    <head>
      <app>
        <lem><supplied resp="#ego" type="emendation">Τμήμα
          πρώτον</supplied></lem>
      </app>
    </head>
    <p>Άνθρακες θερινοὶ ἐν Κραννῶνι. [...]</p>
  </div2>
</div1>

```

## 11.2 TEI Un-numbered Divisions

 As already described on page 7, the un-numbered style of division is the one that is set by default. It is congruous to the general option `divs=ekdosis`.

This style provides a flexible mechanism in which format and presentation are separated from content. It is designed to meet the requirements of classical and literary texts the divisions of which may depend on many different received traditions.

`\ekddiv` `\ekddiv{⟨key-value arguments⟩}` is the unique sectional command provided by `ekdosis`. This command converts the divisions into un-numbered TEI `<div>` elements allowed to nest recursively and takes one mandatory argument in which the following key-value arguments are accepted:—

`type type=⟨name⟩` Default: none  
`type` corresponds to the TEI class `att.typed` and can be used to classify the element in which it is found in any way. Suitable values here can be `book`, `chapter`, `section` and the like.

`n n=⟨value⟩` Default: none  
`n` is meant to provide a number or any kind of label for the division and does not have to be unique in the document.

`head head=⟨name⟩` Default: none  
`head` holds the title of the division and may further contain variant readings.

`barehead barehead=⟨name⟩` Default: none  
`barehead` is supposed to be used to prevent unwanted commands from going into such places as headers, footers and the table of contents.

`depth depth=⟨n⟩` where  $1 \leq n \leq 9$  Default: 1  
As TEI un-numbered divisions are simply `<div>` elements allowed to nest recursively to

indicate their hierarchic depth and `\ekddiv` is an ‘open’  $\LaTeX$  command, `n` is needed to indicate the depth of the division within the hierarchy, the largest being 1 and the smallest being 9.

`toc toc=book|part|chapter|section|subsection|subsubsection|paragraph|subparagraph` Default: not set

If `toc` be set, the title of the division goes into the table of contents at the hierarchic level that is specified as value.

`mark mark=<signpost>` Default: none

*New feature v1.3* `mark` holds the signpost to be emitted as marker for headers and footers. Its value is recalled by `\ekdmark` as described below in [sect. 11.3 on the next page](#).

`\FormatDiv` **Formatting the Titles** By design, `ekdosis` does not format the titles. Instead, depending on what is needed for the edition text, `\FormatDiv{<n>}{<code before>}{<code after>}` is provided to lay out the titles of any hierarchic depth of division. This command takes three mandatory arguments as follows: `<n>`, which is the number referring to the particular depth of division to be formatted and some  $\LaTeX$  commands to go before and after the title itself. The following example illustrates how the titles of the largest division can be printed horizontally centered in a larger size:—

```
\FormatDiv{1}{\begin{center}\Large}\end{center}}
```

To elaborate on the example provided above in [sect. 11.1 on page 58](#), here follows how the first three hierarchical levels could be formatted as un-numbered divisions:—

Listing 12: Divisions of the body text

```
1 % Preamble:
2 \FormatDiv{1}{\begin{center}\Large}\end{center}}
3 \FormatDiv{2}{\begin{center}\large}\end{center}}
4 \FormatDiv{3}{\bfseries}{.}
5
6 % Document:
7 \begin{ekdosis}
8   \ekddiv{
9     head={\Iπποκράτους ἐπιδημιῶν
10       \app{
11         \lem[wit={I,R,H}]{βιβλίον δεῦτερον}
12         \rdg[wit=V]{λόγος β'}},
13     type=book,
14     depth=1,
15     n=II
16   }
17
18   \ekddiv{
19     head={\app{
20       \lem[resp=egomute, post=suppleui,
21         type=emendation]{\supplied{Τμήμα πρώτον}}
22       \rdg[wit=codd, alt=om.]{}},
23     type=section,
24     depth=2,
25     n=II.1
26   }
27
28   \ekddiv{head=1, type=paragraph, depth=3, n=II.1.1}
```

29 Ἄνθρακες θερινοὶ ἐν Κραννῶνι· [...]  
 30 \end{ekdosis}

PDF output:—

1 Ἴπποκράτους ἐπιδημιῶν βιβλίον δεύτερον  
 2 <Τμήμα πρῶτον>  
 3 1. Ἄνθρακες θερινοὶ ἐν Κραννῶνι· [...]

1 βιβλίον δεύτερον IRH] λόγος β' V 2 Τμήμα πρῶτον suppleui] om. codd.

TEI xml output:—

```
<div xml:id="div-hippocrates_1" xml:lang="grc">
  <div type="book" n="II">
    <head>Ἴπποκράτους ἐπιδημιῶν
    <app>
      <lem wit="#I #R #H">βιβλίον δεύτερον</lem>
      <rdg wit="#V">λόγος β'</rdg>
    </app></head>
    <div type="section" n="II.1">
      <head>
        <app>
          <lem resp="#ego" type="emendation">
            <supplied>Τμήμα πρῶτον</supplied>
          </lem>
          <rdg wit="#V #I #R #H" />
        </app>
      </head>
      <div type="paragraph" n="II.1.1">
        <head>1</head>
        <p>Ἄνθρακες θερινοὶ ἐν Κραννῶνι· [...]</p>
      </div>
    </div>
  </div>
</div>
```

### 11.3 Headers and Footers

*New feature v1.3* ekdosis provides a mechanism of its own for emitting header and footer marks. The first operation consists in recording the information to be printed as signpost by means of the mark optional argument of the \ekddiv command described above in [sect. 11.2 on page 59](#).

\ekdmark Once this is done, \ekdmark can be inserted in commands used to make headers and footers where the mark is to be printed.

A common layout is that of headers in which one finds printed in sequence on even pages the page number, then the author's name, and on odd pages the title of the edited text, then the current division, then the page number, like so:—



Even pages:

<code>\thepage</code>	Hippocratis
-----------------------	-------------

Odd pages:

<code>\thepage</code>	Epidemiarum liber II, <code>\ekdmark</code>
-----------------------	---

To return to the example provided by [listing 12 on page 60](#), the mark of the current division would be inserted in the third-level `\ekddiv` command printed at line 28 like so:—

```
28 \ekddiv{head=1, type=paragraph, depth=3, n=II.1.1, mark={1, 1}}
```

The following example finally illustrates how the corresponding headers and footers can be prepared in a straightforward way with the help of the `titleps` package:<sup>71</sup>—

```
% Preamble:
\usepackage{titleps}
\newpagestyle{edition}{
  \sethead[\thepage][Hippocratis][
    ]{Epidemiarum liber II, \ekdmark}{\thepage}
}
% Apply the page style:
\pagestyle{edition}
```

`\endmark` It must be noted that `\ekdmark` is designed to print the first mark that is emitted on a given page and ignore the mark corresponding to any portion of text that may be printed between the top of the page and the point where the first mark is called. `\endmark` is an argument-less command that can be used just at the end of that portion of text to instruct `ekdosis` to print the last-emitted mark of the preceding page instead of the first-emitted mark of the current page.

**Removing Headers and Footers** Pages such as title pages must be printed with empty headers and footers. `ekdosis` must be given control over every item of information inserted in these areas beforehand.

`\ekdprintmark` `\ekdprintmark{<selector>}{<signpost>}` The signposts printed in headers and footers must be passed as second argument of `\ekdprintmark` so that `ekdosis` can remove them on pages where printing them is not desirable. `<selector>` refers to three symbolic letters where the first can be either H or F—for header or footer—, the second E or O—for odd or even—and the third L, C or R—for left, center or right. The example provided above would then read as follows:—

```
% Preamble:
\usepackage{titleps}
\newpagestyle{edition}{
  \sethead[\ekdprintmark{HEL}{\thepage}]
    [\ekdprintmark{HEC}{Hippocratis}]
    []
    {}
    {\ekdprintmark{HOC}{Epidemiarum liber II, \ekdmark}}
    {\ekdprintmark{HOR}{\thepage}}
}
% Apply the page style:
\pagestyle{edition}
```

---

<sup>71</sup> Javier Bezos López, *The Titleps package* (version 2.13) [Page style control] (Oct. 16, 2019), <https://ctan.org/pkg/titleps>.


**“Paired” facing pages** As already described on page 31—the “paired” option—ekdosis can arrange texts on facing pages with every right-hand page number unchanged, so that both facing pages hold the same number, as in the “Budé” series. As a result of this layout, the distinction between left- and right-hand pages does not apply and headers and footers have to be set as if the whole edition text and the translation were intended for single-sided printing. To return to the example just given, the three optional arguments of `\sethead` are inoperative, as are the symbolic letters E and O of `\ekdprintmark`. Both letters must be omitted, which boils down to using the following six selectors, instead of twelve: HL, HC, HR, FL, FC and FR.

`\ekdEOprint` Finally, to make up for commands and arguments designed to set headers and footers on left-hand pages, ekdosis provides `\ekdEOprint`. This command accepts two mandatory, self-evident arguments, like so: `\ekdEOprint{<left-hand mark>}{<right-hand mark>}`.

As an illustration, the example provided above would read as follows for editions laid out on “paired” facing pages:—

```
% Preamble:
\usepackage{titleps}
\newpagestyle{edition}{
  \sethead{\ekdprintmark{HL}{\ekdEOprint{\thepage}}}{
    {\ekdprintmark{HC}{\ekdEOprint{Hippocratis}}{
      Epidemiarum liber II, \ekdmark}}
    {\ekdprintmark{HR}{\ekdEOprint{}{\thepage}}}
  }
% Apply the page style:
\pagestyle{edition}
```

**“Mirrored” paired facing pages** The “paired” option just described can give surprising results at first sight, in which facing pages are arranged by succeeding pairs of left- or right-hand pages. This is certainly not the desired layout for duplex printing. However, this is not a redhibitory inconvenience for professional printers can easily carry out the necessary corrections by applying offsets to the horizontal margins. That being said, one may also wish to have these horizontal offsets applied from the outset in the PDF output that is produced by L<sup>A</sup>T<sub>E</sub>X.

 What follows assumes that the critical edition is printed on facing pages and starts on a left-hand page. Furthermore, it must be noted that the `paired` option must be either unset or set to `false`.

`\setpairedpage` `\setpairedpage` is provided to set mirrored paired facing pages. This command takes no argument and is meant to be issued in commands used to set headers and footers on both sides, *before any call* of the commands `\thepage` or `\ekdmark`, like so:—

```
1 % Preamble:
2 \usepackage{titleps}
3 \newpagestyle{edition}{
4   \sethead[\setpairedpage\ekdprintmark{HEL}{\thepage}]
5     [\ekdprintmark{HEC}{Hippocratis}]
6     []
7   {}
8   {\setpairedpage\ekdprintmark{HOC}{Epidemiarum liber II,\
9     \ekdmark}}
10  {\ekdprintmark{HOR}{\thepage}}
11 }
```

```
12 % Apply the page style:
13 \pagestyle{edition}
```

REM. 1 As can be seen from ll. 4 and 8, `\setpairedpage` must be issued *before* `\ekdprintmark` so that it can be processed on pages where headers and footers are removed by `\ekdnohfmarks` described below.

REM. 2 Additionally, ll. 8–10 show that `\setpairedpage` precedes both `\ekdmark` and `\thepage`.

REM. 3 If there be no `\thepage` in headers or footers, `\setpairedpage` must still be issued once on each side.

`\setpairedpagenum` `\setpairedpagenum{number}` can be used just ahead of the alignment environment to set the number of the first left-hand paired page to follow. The first paired page number is initially set to 1.

`\resetpagenumber` Finally, `\resetpagenumber` *must* be used right out of the alignment environment so that any numbering error on the page that follows the edition text can be corrected. See below [listing 13](#), l. 16.

A schematic illustration of this technique follows:—

Listing 13: “Mirrored” paired facing pages

```
1 \mainmatter
2 \part{Edition text}
3 \cleartoevenpage % this needs the 'nextpage' package.
4 \pagestyle{edition} % apply the style where 'mirrored' paired facing
5 % pages are defined as illustrated above.
6 \begin{alignment}
7   \begin{edition}
8     ...
9   \end{edition}
10  \begin{translation}
11    ...
12  \end{translation}
13 \end{alignment}
14 \backmatter % or \cleartooddpage provided by the 'nextpage'
15 % package.
16 \resetpagenumber % this command is always required.
17 \pagestyle{back} % apply a new page style.
```

`\ekdnohfmarks` **Removing or Resetting Headers and Footers** Once all signposts are marked with `\ekdprintmark`, `\ekdnohfmarks` can be used at any point of the document with the same effect as the L<sup>A</sup>T<sub>E</sub>X standard command `\thispagestyle{empty}`.

`\ekdresethfmarks` Finally, `\ekdresethfmarks` is provided in rare cases when it is needed to reset headers and footers to their original, viz. printable state.

## 12 The Tricks of the Trade

As the `.tex` source file is compiled, `ekdosis` has to compute a tremendous amount of data. Most of this work is performed by Lua functions. An edition text narrowed down to a single page needs to be compiled at least three times. On the first run, the apparatus criticus does not show. Instead, `ekdosis` produces an auxiliary file named `\jobname.ekd` in which all the entries of the apparatus criticus are collected. Then, on the second run a test is performed on this auxiliary file to determine whether there are entries—and if so, which ones—to be printed on the current page. At the same time, references to the line numbers are updated if necessary. Finally, on the third run, the apparatus criticus is printed.

Of course, every change made to the input may similarly require Lua $\LaTeX$  to be run three more times to get everything to the right place with the right numbers.

## 12.1 The Oscillating Problem

In some instances, notably when on a given page entries are very abundant in number, specifically when the edition text is getting close to the bottom of the page, `ekdosis` may oscillate indefinitely between different sets of page decisions without being able to settle down. The condition may be typically illustrated as follows: after Lua $\LaTeX$  has been run, an entry is attached to the last line of the page. As said above, this entry does not show yet. But when it does, should it result in an additional line being printed in the apparatus criticus, the last line of the edition text—the one the entry was previously attached to—goes to the next page. As a result, this entry also moves to the next page with the line it belongs to. This point is literally critical, because unless a `\pagebreak` is inserted just here so as to keep the contentious line on the next page, `ekdosis` enters a vicious circle from which it cannot escape, not to mention that right entries with right line numbers cannot come on pages that follow a wrong page either.

An alert reader may have guessed that inserting a `\pagebreak` is a good way to get out of the vicious circle. And surely, if only a few pages be at stake, this is the way to go. However, `\pagebreak` commands should only be inserted when the whole edition text is ready for any substantial change in the preceding pages may result in pages that break just after they begin.

`\ekdpb` **Conditional page breaks** `\ekdpb[⟨page no⟩]{⟨line no⟩} \ekdpb*{} \ekdpb*`

`\ekdpb*` One way to avoid this inconvenience is to use `\ekdpb` instead of the standard `\pagebreak` command provided by  $\LaTeX$  to insert conditional page breaks. `\ekdpb` takes as mandatory argument the line number, as it is printed in the margin, where the page break should take place. An optional argument allows to further specify the page number where the page break should occur. The value that is expected is the page number as it is printed—e.g. an Arabic, Roman or alphanumeric number. If the specified conditions be not met, then the page break is not triggered. Finally, the “starred” version of this command forces the page break, irrespective of the values specified as page or line numbers. Unlike `\ekdpb`, which requires the lines to be numbered, `\ekdpb*` is allowed at any point of the document: as `\ekdpb*` disregards the number given as argument, it is equivalent to the standard  $\LaTeX$  `\pagebreak` command. Yet it can be used instead of the latter to have marks further printed in the margins so as to spot with a fleeting glance the locations where induced page breaks occur.<sup>72</sup>

**Using `maxentries`** Another way—should the edition text fall into the vicious circle too often—is to limit the number of entries per page that all layers of critical notes taken together or a given layer of apparatus criticus may accept as described above on pages 35 and 39. As a result, `ekdosis` will take care of inserting automatic breakpoints between pages whenever the number of entries on a given page reaches the value set as `maxentries`.

`maxentries` must not be too small: otherwise offensive to look at vertical spaces may come between the edition text and the apparatus criticus. Conversely, `maxentries` must not be too big: otherwise, should entries overflow on a given page, the edition text and the apparatus criticus may clash again. As said above, a couple of clashes can be managed with a couple of manually inserted page breaks. But if there be too many of them, it is a good indication that the selected value of `maxentries` is too high.

---

72. This requires the `showpagebreaks` option to be set to `true` as described above on page 7.

Complex edition texts do have a magic number. An advisable way to figure it out would be to start from a sample of only a few pages, selected as evidence for the complexity of the whole. As only a few pages would need to be compiled, the magic number should emerge quite rapidly.

`\addentries` **Adding and Removing Entries** `\addentries[⟨layer⟩]{⟨n⟩}`  
*New feature v1.1* If `maxentries` be set for a given layer of critical notes, `\addentries[⟨layer⟩]{⟨n⟩}`, where `⟨n⟩` is an integer, can be used to add `⟨n⟩` to—or remove it from if `⟨n⟩` be negative—the number of accepted entries on the current page. `\addentries` operates on the default layer of notes, but any other declared layer can be specified in the optional argument of the command.



Of course, `\addentries` must be issued before the number of entries on a given page has reached the value set as `maxentries`.

Once a sensible value for `maxentries` has been found, `\addentries` can further be used with a positive integer to allow for more entries and more lines on some pages so that offending vertical spaces are decreased. Conversely, `\addentries` with a negative integer will remove entries on pages where there are too many of them and `ekdosis` still oscillates between different sets of page decisions.

*New feature v1.3* **The `fitapp` Global Option** The rationale of this option is discussed above (see (d) on page 6). As this mechanism has the characters of the apparatus block scaled down to allow for more entries once a predefined height has been reached, `ekdosis` should settle down in most of the contentious cases.<sup>73</sup> However, it is advisable to use `fitapp` conjointly with `maxentries` to prevent the number of allowed entries from being too high, which would result in the characters being too small or even illegible.

## 12.2 *Variae Quaestiones*

This section is about issues that are not strictly speaking part of the documentation of `ekdosis` but may nevertheless circumstantially arise.

**Superfluous Dots** As said above on page 36, it is customary in some editions to have a full stop printed at the end of the apparatus criticus. `ekdosis` provides specific commands to achieve this in a straightforward way, such as `\SetEndApparatus` and the `ehook` optional argument of `\SetApparatus` and `\DeclareApparatus`.<sup>74</sup> However, if the last word of the apparatus criticus on a given page be an abbreviation followed by a dot, such a setting will have two dots printed at the end of the apparatus instead of one. The solution is to define a command to have a dot printed only if not followed by a dot, and append this command to the abbreviated form of the word, like so:—

```

1 % Preamble:
2 \usepackage{xspace}
3 \usepackage{ekdosis}
4
5 \makeatletter
6 \newcommand{\ekddot}{%
7   \ltx@ifnextchar@nospace{.}{\xspace}{.\xspace}}
8 \makeatother

```

<sup>73</sup> That is, cases that arise *after* the predefined height has been reached.

<sup>74</sup> See above on pages 36–38.

```

9
10 \DeclareApparatus{default}[ehook=.]
11 \DeclareScholar{Erm}{Erm\ekddot}

```

- REM. 1 Line 2: The `xspace` package is needed for `\xspace` is used by the `\ekddot` command that is defined at l. 6.
- REM. 2 Line 7: `\ltx@ifnextchar@nospace` is part of the `ltxcmds` package which is loaded by `ekdosis`. As this command uses a private control sequence, it must be found within `\makeatletter ... \makeatother`.
- REM. 3 Line 10: `\ekddot` will only work with multiple-layer apparatus criticus. Therefore, `\DeclareApparatus` must be used even if only one layer of critical notes be needed.

**Backup of Essential Files** Each time the `.tex` source file is compiled, `ekdosis` reads the `.aux` corresponding L<sup>A</sup>T<sub>E</sub>X auxiliary file and its own `.ekd` auxiliary file so as to process labels and collect entries of the apparatus criticus. If for whatever reason—e.g. some unknown command has been inserted—the compilation be frozen and so must be aborted, it may happen that most of the edition text has to be reconstructed page after page. For large and complex editions, this makes advisable to have current versions of those files backed up each time a new compilation begins, which can be achieved by inserting the following lines before the line that loads the document class:—

```

\RequirePackage{verbatimcopy}
\IfFileExists{\jobname.aux}{%
  \OldVerbatimCopy{\jobname.aux}{\jobname.aux.bak}}{}
\IfFileExists{\jobname.ekd}{%
  \OldVerbatimCopy{\jobname.ekd}{\jobname.ekd.bak}}{}
\documentclass{book}

```

This way, both `.aux` and `.ekd` files can be recovered from `.aux.bak` and `.ekd.bak` just after the compilation has been aborted. Should this be needed, one must proceed carefully as follows:—

- (a) Just after the compilation has been aborted, move both `aux.bak` and `ekd.bak` files to a safe place.
- (b) Remove or correct the offending command or lines that broke the compilation and make sure that the issue is solved.
- (c) Restore the `.aux` and `.ekd` files from `aux.bak` and `ekd.bak` and resume work where it was left off.

## 13 TEI xml Output

Several examples of TEI xml output have been provided hitherto. Before proceeding, the reader is invited to return to every one of them. In this respect, it may be of interest to review carefully the excerpt of Caesar’s *Gallic War* of which the L<sup>A</sup>T<sub>E</sub>X source file and its corresponding TEI xml output are printed in full below in [sect. 17 on page 90](#). Once `ekdosis` has been instructed to convert the edition text into TEI xml (l. 11), the preamble of this file shows how to set languages and fonts to be used in the document (ll. 2–6), format the titles (l. 16) and lay out the alignment of an edition text associated with two translations (ll. 18–25) in modern languages. Furthermore, it shows how information related to each language (Latin, English and French) is to be found in two different places, namely for TEI xml output (ll. 21–3) and for PDF output through L<sup>A</sup>T<sub>E</sub>X (ll. 27–9). Finally, it provides examples of declaring witnesses, hands and shorthands (ll. 31–60). As to the document itself, it shows how to lay out a conspectus siglorum in a table (ll. 64–80), before giving

detailed examples of how the edition text is entered (ll. 85–101) and sectional commands provided by ekdosis are used (ll. 86, 103 and 110).<sup>75</sup>

### 13.1 Requesting TEI xml Output

TEI xml output is requested by means of the `teiexport` global option as described above on page 7. Once instructed to output TEI, ekdosis converts and exports in sequence the contents of ekdosis environments (see above [sect. 2.5 on page 13](#)). As regards the contents of alignment environments (see above [sect. 5 on page 27](#)), ekdosis first collates the contents of the environments that have been declared as values of the `texts` optional argument of `alignment` or `\SetAlignment`,<sup>76</sup> then places each of the corresponding TEI xml outputs within distinct `<div>` elements named after the declared environments themselves. For example, to return to Caesar’s text, the Latin edition text is found between a `\begin{latin} ... \end{latin}` environment (see the `.tex` source file, [sect. 17.1 on page 90](#), ll. 85–101) which is declared at l. 21. Then, the corresponding xml output is found within a `<div>` element, the `xml:id` of which has been given by ekdosis the value `div-latin_1` (see [sect. 17.2 on page 93](#), ll. 176–200).

`\SetTEIFilename` **TEI File Name** `\SetTEIFilename{(basename)}` is a preamble-only command. It can be used to set the base name of the TEI xml output file, to which the suffix `.xml` is appended. By default, the base name is `\jobname-tei`.

### 13.2 General Principles

**Validation of the TEI xml Structure** The reference tool that the author relies on is that provided by the *TEI by Example Project*.<sup>77</sup> As for ekdosis, it is designed to produce on request, in addition to an edition in print, a TEI xml-compliant output file. That said, one must keep in mind that the L<sup>A</sup>T<sub>E</sub>X packages that are part of T<sub>E</sub>XLive can be counted in thousands, and the commands they provide in tens of thousands. There may even be grounds in asserting that the possibilities offered by T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X quite exceed what can be afforded by TEI xml. On another hand, many L<sup>A</sup>T<sub>E</sub>X commands make no sense in TEI. Therefore, a sensible choice is to keep them out of the environments the contents of which are to be translated into xml elements, as will be illustrated by the following.

Converting a L<sup>A</sup>T<sub>E</sub>X document into TEI xml can be quite an intricate business. In many cases, however, L<sup>A</sup>T<sub>E</sub>X strings are found within environments or groups that are easy to convert into TEI equivalents: unless instructed otherwise, whether such groups are delimited by opening and closing braces or by explicit `\begin ... \end` commands, ekdosis translates them into xml so that for example `\emph{word}` and `\begin{quote} <quoted words> \end{quote}` become `<emph> <word> </emph>` and `<quote> <quoted words> </quote>` respectively.

But L<sup>A</sup>T<sub>E</sub>X does not place everything into groups or environments. To take here but a few examples, sectional divisions are marked in L<sup>A</sup>T<sub>E</sub>X with “open” commands such as `\chapter` or `\section` with no clear indication where the closure of divisions occurs, contrary to TEI xml markup with numbered or un-numbered `<div>` elements allowed to nest recursively. As regards running paragraphs of text, the situation is even worse than in the latter case, as the following simple example shows:—

---

<sup>75</sup> The PDF output is available as a [separate file](#).

<sup>76</sup> See above [sect. 5.1 on page 30](#).

<sup>77</sup> Ron Van den Branden, Melissa Terras, and Edward Vanhoutte, “TEI by Example,” <http://www.teibyexample.org>, accessed Aug. 4, 2020. The TEI validator is here: <http://teibyexample.com/xquery/TBValidateor.xq>.



```

1  \begin{document}
2  \begin{ekdosis}
3  ...
4
5  ... These are the final words of some section in the body text.
6
7  \section{New Section}
8
9  Here is how some new section begins...
10
11 ... Final words.
12 \section{Other Section}
13 Opening words of the section...
14
15 ... Final words
16
17 \section{Other Section}
18 Opening words...
19
20 ... Final words.
21 \end{ekdosis}
22 \end{document}

```

Obviously, construing this L<sup>A</sup>T<sub>E</sub>X source file into TEI xml is a fairly complex task. For example, line 6 only closes a paragraph for line 7 opens a division (hence `</p><div1>`), line 8 only opens a paragraph just after the heading of the section (hence `</head><p>`) while line 14 both closes the foregoing paragraph and opens a new one (hence `</p><p>`), contrary to line 16 which both closes a paragraph and a sectional division (hence `</p></div1>`), not to mention lines 20–1, where notwithstanding the absence of blank line or any other indication, `</p></div1></body></text></TEI>` is needed.

ekdosis has been designed to implement this task through Lua functions which involve string matching (both forward and reverse matching) and recursions.

`\SetTEIxmlExport` **TEI xml Export Settings** `\SetTEIxmlExport{csv list of options}` can be used in the preamble or at any point of the document, except inside environments set to receive an apparatus criticus, namely the `ekdosis` environment or any other similar environment declared by means of the `alignment` environment or `\SetAlignment`.<sup>78</sup> At the time of writing, there is only one option, as follows:—

`autopar` `autopar=true|false` Default: true


The algorithm described above applies for edition texts composed in running paragraphs or lines of poetry, but it may fail to produce a valid TEI xml output with other arrangements, such as performance texts or transcriptions of speech for which the TEI Guidelines define specific rules. `autopar=false` instructs `ekdosis` to ignore blank lines in the `.tex` source file as markers for paragraph boundaries. As a result, each paragraph of the edition text must be found within an environment associated with the `xml` element `<p>`, such as `ekdpar` or any other environment declared as such by means of `\EnvtoTEI` described below in [sect. 13.4 on page 72](#). A typical use case of `autopar=false` is provided below in [sect. 13.6 on page 75](#).


`ekdpar` (*env.*) `\begin{ekdpar} ... \end{ekdpar}` is a simple environment that does nothing but insert `\par` primitives. It can be used to instruct `ekdosis` to place paragraphs within `<p>` elements when `autopar` has been set to `false` by means of `\SetTEIxmlExport` described above.

<sup>78</sup>. See above [sect. 5.1 on page 30](#).



**The `xml:id` Attribute** As a general rule, the `xml:id` global attribute must be unique for the element that bears the attribute. Furthermore, it must begin with a letter or an underscore and contain no characters other than letters of the Latin alphabet—from a to z, either upper or lower case—digits, hyphens, underscores and full stops. `ekdosis` issues a warning when it finds that any *unique id* of *unique label* expected in the first argument of `\DeclareWitness`, `\DeclareHand`, `\DeclareSource` or `\DeclareScholar` is not unique or breaks the rules just described, but does not prevent the `.tex` source file from compiling. Instead, it prints the string `<??>` in place of the expected formatted siglum so that the error in the `.tex` source file can be easily spotted and corrected.

 As the *unique id* declared with `\DeclareShorthand` is not to be exported in the TEI `xml` output file, `ekdosis` checks neither its uniqueness nor its validness.

 It must be noted that  $\LaTeX$  labels that are provided in commands such as `\label`, `\cite` and the like must also be unique in the document. As  $\LaTeX$  will issue warnings if duplicates be found, `\ekdosis` does not check their uniqueness but will issue warnings if such labels contain invalid strings.

### 13.3 Routine $\LaTeX$ Commands and Environments

The list of  $\LaTeX$  commands known by `ekdosis` at the time of writing follows. To this list must be added the  $\LaTeX$  standard commands that are used for sectional divisions as described above in [sect. 11.1 on page 58](#) and most of the commands provided by the `arabxat` and `icite`<sup>79</sup> packages. Standard citation commands are also supported as will be described below in [sect. 13.8 on page 79](#):—

$\LaTeX$ command	TEI <code>xml</code> element
<code>\textsuperscript{}</code>	<code>&lt;hi rend="sup"&gt;&lt;/hi&gt;</code>
<code>\textsubscript{}</code>	<code>&lt;hi rend="sub"&gt;&lt;/hi&gt;</code>
<code>\textbf{}</code>	<code>&lt;hi rend="bold"&gt;&lt;/hi&gt;</code>
<code>\textit{}</code>	<code>&lt;hi rend="italic"&gt;&lt;/hi&gt;</code>
<code>\textsc{}</code>	<code>&lt;hi rend="smallcaps"&gt;&lt;/hi&gt;</code>
<code>\textsf{}</code>	<code>&lt;hi rend="sf"&gt;&lt;/hi&gt;</code>
<code>\footnote{}</code>	<code>&lt;note place="bottom"&gt;&lt;/note&gt;</code>
<code>\marginpar{}</code>	<code>&lt;note place="margin"&gt;&lt;/note&gt;</code>
<code>\enquote{*}{}</code>	<code>&lt;quote&gt;&lt;/quote&gt;</code>
<code>\label{label}</code>	<code>&lt;anchor xml:id="label"/&gt;</code>
<code>\linelabel{label}</code>	<code>&lt;anchor xml:id="label"/&gt;</code>
<code>\ref{label}</code>	<code>&lt;ptr ="#label"/&gt;</code>
<code>\pageref{label}</code>	<code>&lt;ptr ="#label"/&gt;</code>
<code>\vref{label}</code>	<code>&lt;ptr ="#label"/&gt;</code>
<code>\vpageref{label}</code>	<code>&lt;ptr ="#label"/&gt;</code>
<code>\pagebreak&lt;[&lt;1-4&gt;]&gt;</code>	no output
<code>\mbox{&lt;text&gt;}</code>	<code>&lt;text&gt;</code>
From the <code>marginnote</code> <sup>80</sup> package:	
<code>\marginnote{}</code>	<code>&lt;note place="margin"&gt;&lt;/note&gt;</code>
From the <code>extdash</code> <sup>81</sup> package:	
<code>\---</code> or <code>\===</code>	—

<sup>79</sup> Robert Alessi, *The Icite package* (version 1.3a) [Indices locorum citatorum] (Mar. 5, 2020), <http://ctan.org/pkg/icite>.

L <sup>A</sup> T <sub>E</sub> X command	TEI xml element
<code>\--</code> or <code>\==</code>	–
<code>\-/</code> or <code>\=/</code>	-

As for environments:—

L <sup>A</sup> T <sub>E</sub> X environment	TEI xml element
<code>flushright</code>	<code>&lt;p rend="align(right)"&gt;&lt;/p&gt;</code>
<code>flushleft</code>	<code>&lt;p rend="align(left)"&gt;&lt;/p&gt;</code>
<code>center</code>	<code>&lt;p rend="align(center)"&gt;&lt;/p&gt;</code>
<code>quotation</code>	<code>&lt;quote&gt;&lt;/quote&gt;</code>
<code>quoting</code>	<code>&lt;quote&gt;&lt;/quote&gt;</code>
<code>verse</code>	<code>&lt;lg&gt;&lt;/lg&gt;</code>

Regarding other, very frequently used commands or environments, some do not need to be inserted in the translation tables: as already said above, `ekdosis` converts by default the original names of these into xml elements. For instance, `\emph{}` and `\begin{quote}` ... `\end{quote}` will result in `<emph></emph>` and `<quote></quote>` respectively.

For the same simple reason, should one wish to have words within a TEI xml element that does not have any L<sup>A</sup>T<sub>E</sub>X equivalent, all is needed is to define an inoperative L<sup>A</sup>T<sub>E</sub>X command named after the TEI element, like so:—

```
% Preamble:
\newcommand{\mentioned}[1]{#1}

% Document:

Our usage corresponds to the \mentioned{aggregate} of many
mathematical writings and to the sense of \mentioned{class} found in
older logical writings.
```

TEI xml output:—

```
<p>Our usage corresponds to the <mentioned>aggregate</mentioned> of
many mathematical writings and to the sense of
<mentioned>class</mentioned> found in older logical writings.</p>
```

Of course, it is also possible to have the “mentioned” words printed in a different font family:—

```
\newcommand{\mentioned}[1]{\textsf{#1}}
```

This command will print them in a sans serif font family, with the exact same TEI xml output as above.

80. Markus Kohm, *The Marginnote package* (version 1.4b) [Notes in the margin, even where `\marginpar` fails] (Aug. 9, 2018), <https://komascript.de/marginnote>.

81. Alexander I. Rozhenko, *The Extdash package* (version 1.3) [A range of dash commands for compound words] (June 24, 2018), <http://www.ctan.org/pkg/extdash>.

## 13.4 Processing New Commands or Environments

The following three commands are provided to instruct ekdosis how to convert unknown or unusual (L)T<sub>E</sub>X commands or environments into TEI xml equivalents.

```
\TeXtoTEI \TeXtoTEI{<cname>}{<TEI element>}[<TEI attribute(s)>]
```

`\TeXtoTEI` takes two mandatory arguments and one optional argument, namely: the control sequence name to be converted, the TEI element it is to be converted into and any additional xml attributes to be appended to the opening TEI element. For example, the `\sidenote` command that is provided by the `sidenotes` package can be processed like so:—

```
% Preamble:
\TeXtoTEI{sidenote}{note}[place="margin"]

% Document:
\begin{ekdosis}
  \begin{ekdverse}
    The self-same moment I could pray;\sidenote{The spell begins to
      break}\footnote{The turning point of the poem...}
  \end{ekdverse}
\end{ekdosis}
```

TEI xml output:—

```
<lg>
<l>The self-same moment I could pray;
<note place="margin">The spell begins to break</note>
<note place="bottom">The turning point of the
poem...</note></l>
</lg>
```

Even more subtly, provided that the code `#STC` points to some more information identifying the agency concerned:<sup>82</sup>—

```
% Preamble:
\usepackage{sidenotes}
\usepackage[telexport=tidy]{ekdosis}

\TeXtoTEI{sidenote}{note}[place="margin"]

\newcommand{\STCsnote}[1]{\sidenote{#1}}
\TeXtoTEI{STCsnote}{note}[place="margin" resp="#STC"]

% Document:
\begin{ekdosis}
  \begin{ekdverse}
    The self-same moment I could pray;\STCsnote{The spell begins to
      break}\footnote{The turning point of the poem...}
  \end{ekdverse}
\end{ekdosis}
```

TEI xml output:—

---

<sup>82</sup>. At the time of writing, ‘sources’ can be declared with `\DeclareSource` as described above on page 10. Then the unique identifier used in the first argument of this command can point to the list of references inserted by ekdosis in the back matter section of the TEI output file. See below [sect. 13.7 on page 76](#) for more information on how to do this. Scholars can also be referred to as individuals by means of the `\DeclareScholar` command. See above on page 10.

```

<lg>
  <l>The self-same moment I could pray;
  <note place="margin" resp="#STC">The spell begins to
  break</note>
  <note place="bottom">The turning point of the
  poem...</note></l>
</lg>

```

`\EnvtoTEI` `\EnvtoTEI{*}{<env name>}{<TEI element>}[<TEI attribute(s)>]`

`\EnvtoTEI*` `\EnvtoTEI` instructs `ekdosis` how to convert  $\LaTeX$  environments into TEI `xml` equivalents. It takes two mandatory arguments and one optional argument, namely the name of the  $\LaTeX$  environment to be converted, the TEI element it is to be converted into and any additional attributes to be appended to the TEI opening element. `\EnvtoTEI*` is restricted to TEI elements that must never appear within `<p>` elements, such as `<p>` itself, `<div>`, `<lg>` and the like. The following example illustrates how `\EnvtoTEI` can be used conjointly with `babel` to convey information about the languages used from  $\LaTeX$  to TEI:—

```

% Preamble:
% Use babel and babeltags:
\usepackage[greek.ancient, english]{babel}
\babeltags{ancientgreek = greek}

\EnvtoTEI{ancientgreek}{p}[xml:lang="grc"]

% Document:
\begin{ekdosis}
  \begin{ancientgreek}
    περί πολλοῦ ἄν ποιησαίμην, ὃ ἄνδρες, τὸ τοιούτους ὑμᾶς ἐμοὶ
    δικαστὰς περὶ τούτου τοῦ πράγματος γενέσθαι, οἷοίπερ ἄν ὑμῖν
    αὐτοῖς εἴητε τοιαῦτα πεπονθότες...
  \end{ancientgreek}
\end{ekdosis}

```

TEI `xml` output:—

```

<p xml:lang="grc">περί πολλοῦ ἄν ποιησαίμην, ὃ ἄνδρες, τὸ
τοιούτους ὑμᾶς ἐμοὶ δικαστὰς περὶ τούτου τοῦ πράγματος
γενέσθαι, οἷοίπερ ἄν ὑμῖν αὐτοῖς εἴητε τοιαῦτα πεπονθότες...</p>

```

`\TeXtoTEIPat` `\TeXtoTEIPat{<TEX pattern>}{<TEI pattern>}`

Finally, this more flexible—and more delicate to handle—command uses pattern matching to instruct `ekdosis` how to convert  $\LaTeX$  commands into TEI equivalents. In the first mandatory argument, strings to be captured are marked in sequence with numbers prefixed by `#`, like so: `#1`, `#2`, `#3` and so forth. Then, in the second mandatory argument, the strings captured are inserted where each of them is expected in the TEI element.

⚠ If the entire string to be captured be enclosed in square or curly brackets, it is advisable to use `@bn` (for curly brackets) or `@sn` (for square brackets) instead of `#n`, where `n` is the number that is expected in the sequence. This will prevent any brackets that may be found in the captured string from being interpreted.

⚠ Strings must be entered exactly as `ekdosis` will find them as the `.tex` source file is compiled. Specifically, *control sequences*, namely the coded commands immediately preceded by `\` are always found followed by a space. For instance, `\emph{}` will be seen and processed by `ekdosis` as `\emph_{}{}`.

The following example illustrates how `ekdosis` can be instructed to process the `\textcolor{<color>}{<text>}` command:—

```
1 \TeXtoTEIPat{\textcolor_{#1}@b2}{<hi rend="#1">@b2</hi>}
2
3 Sample text with a \textcolor{red}{word} in red.
```

REM. As can be seen from l. 1, it is safe to use `#1` for the first string for color names are naturally formed of letters without braces. However, `@b2` is preferable to capture the whole second argument of `\textcolor` for it may contain words within braces.

```
<p>Sample text with a
<hi rend="red">word</hi>in red.</p>
```

`\getTEIxmlid` **Inserting xml:ids in TEI Patterns** In some instances, it may be useful to retrieve the `xml` formatted pointers corresponding to unique identifiers declared in commands such as `\DeclareWitness`<sup>83</sup> and the like so as to insert them in the second argument—(*TEI pattern*)—of `\TeXtoTEIPat`. As an illustration, what follows creates a new  $\text{\LaTeX}$  command `\witStart` which prints nothing in the PDF output but inserts `<witStart/>` elements in the TEI `xml` file. This command accepts an optional argument to make clear which are the fragmentary witnesses involved:—

```
1 \NewDocumentCommand{\witStart}{0}{\ignorespaces}
2 \TeXtoTEIPat{\witStart [wit=@b1]}{<witStart wit="\getTEIxmlid{@b1}"/>}
3 \TeXtoTEI{\witStart}{witStart}
```

REM. 1 On the use of `\NewDocumentCommand` to create new commands, the reader is invited to refer to the documentation of the `xparse` package.<sup>84</sup>

REM. 2 As can be seen from l. 2, `\getTEIxmlid` is used to process the `csv`-list of unique identifiers that matches the string captured between the braces after `wit=`, as implied by `@b1`. As a consequence, `\witStart []` is expected as follows: `\witStart [wit={X,Y}]`. Braces must exist for pattern matching.

REM. 3 Finally, `\TeXtoTEI{\witStart}{witStart}` is used so that the argument-less form of the command can be processed as well.

### 13.5 Inserting Code in the TEI xml Output File

It may be needed to insert code in the TEI `xml` output file only, for example when clear enough information is written in the apparatus criticus by means of such optional arguments as `pre`, `post`, `prewit` or `postwit` that are not processed for TEI `xml` output.<sup>85</sup>

`\teidirect` `\teidirect[<xml attributes>]{<xml element>}{<code>}`  
*New feature v1.3*

Two mandatory arguments are expected by `\teidirect`, namely the TEI `xml` element followed by the contents to be found in the output file between the opening and closing tags. Additionally, attribute-value pairs to be found inside the start-tag of the element can be specified in the optional argument of the command.

`\teidirectE` `\teidirectE[<xml attributes>]{<xml element>}`  
*New feature v1.5* `\teidirectE`—where E stands for “empty element”—is strictly equivalent to `\teidirect [<xml attributes>]{<xml element>}{}`. Compared to the latter command, it only eliminates the need to insert the ending pair of empty braces.

An example follows:

<sup>83</sup>. See above on page 8.

<sup>84</sup>. The LaTeX Team, *The Xparse package* [A generic document command parser] (Jan. 12, 2022), <https://ctan.org/pkg/xparse>.

<sup>85</sup>. See above [sect. 2.5 on page 13](#).

```

1 % Preamble:
2 \DeclareWitness{GalE1.M}{Gal.E1(M)}{\emph{Monacensis Gr.}
3 231}[origDate=s. XVI]
4 \DeclareWitness{GalE1.Q}{Gal.E1(Q)}{\emph{Parisinus Gr.}
5 2174}[origDate=s. XIV]
6 \DeclareShorthand{GalE1.M.Q}{Gal.E1(MQ)}{GalE1.M,GalE1.Q}
7
8 % Document:
9 ἐν \app{
10 \lem[wit=codd]{καύμασιν}
11 \rdg[wit=GalE1.M.Q, postwit=\unskip(23.16)]{καύματι}
12 \teidirect{note}{p. 23, l. 16 Wenkebach}}

```

REM. 1 GalE1.M.Q (l. 11) has been defined as a shorthand to denote the agreement of two otherwise defined manuscripts by means of \DeclareWitness: GalE1.M and GalE1.Q. (See ll. 2–6.)

REM. 2 The postwit optional argument has been used to further specify the location where this variant reading can be found in the critical edition of Galen’s Commentary on Hippocrates’ *Epidemics*, Book 1 (l. 11). But as the effect of postwit is limited to the PDF output, \teidirect has been used to convey this item of information to the TEI xml output file (l. 12).

PDF output:—

```

1 ἐν καύμασιν

```

---

I καύμασιν codd.] καύματι GalE1(MQ)(23.16)

TEI xml output:—

```

1 <p xml:lang="grc">ἐν
2 <app>
3 <lem wit="#V #I #R #H">καύμασιν</lem>
4 <rdg wit="#GalE1.M #GalE1.Q">καύματι</rdg>
5 <note>p. 23, l. 16 Wenkebach</note>
6 </app></p>

```

## 13.6 Specific TEI Modules

The following example illustrates how ekdosis can be adapted in a straightforward way to modules provided by the TEI for encoding specific texts such as transcriptions of speech.<sup>86</sup> The technique applied below uses \EnvtoTEI conjointly with \SetTEIxmlExport{autopar=false} described above on page 69:—

```

1 % Preamble:
2 \newenvironment{speech}{\par}{\par}
3 \newcommand{\speaker}[1]{\textbf{#1}\par}
4 \EnvtoTEI{speech}{sp}
5
6 \SetTEIxmlExport{autopar=false}
7
8 % Document:
9 \begin{ekdosis}
10 \begin{speech}

```

<sup>86</sup>. See <https://tei-c.org/release/doc/tei-p5-doc/en/html/TS.html>.

```

11 \speaker{Σωκράτης}
12 \begin{ekdpar}
13 κατέβην χθές εἰς Πειραιᾶ μετὰ Γλαύκωνος τοῦ Ἀρίστωνος
14 προσευξόμενός τε τῇ θεῷ καὶ ἅμα τὴν ἑορτὴν βουλόμενος θεάσασθαι
15 τίνα τρόπον ποιήσουσιν ἄτε νῦν πρῶτον ἄγοντες. καλὴ μὲν οὖν μοι
16 καὶ ἡ τῶν ἐπιχωρίων πομπὴ ἔδοξεν εἶναι, οὐ μέντοι ἦττον ἐφαίνετο
17 πρέπειν ἢν οἱ θοῤᾳκες ἔπεμπον.
18 \end{ekdpar}
19 \end{speech}
20 \end{ekdosis}

```

REM. 1 Lines 2–3 define a basic environment meant to contain individual speeches and a command to hold the name of the speaker. This name is printed in bold face and followed by a new paragraph in the PDF output.

REM. 2 Line 4 instructs ekdosis to convert speech L<sup>A</sup>T<sub>E</sub>X environments into <sp> TEI xml elements.

REM. 3 Line 6 disables the autopar algorithm that ekdosis provides by default for running paragraphs of text. As a consequence, ekdpar is used to mark the paragraphs.

PDF output:—

```

1 Σωκράτης
2 κατέβην χθές εἰς Πειραιᾶ μετὰ Γλαύκωνος τοῦ Ἀρίστωνος προσευξόμενός τε τῇ θεῷ καὶ
3 ἅμα τὴν ἑορτὴν βουλόμενος θεάσασθαι τίνα τρόπον ποιήσουσιν ἄτε νῦν πρῶτον ἄγοντες.
4 καλὴ μὲν οὖν μοι καὶ ἡ τῶν ἐπιχωρίων πομπὴ ἔδοξεν εἶναι, οὐ μέντοι ἦττον ἐφαίνετο πρέπειν
5 ἢν οἱ Θοῤᾳκες ἔπεμπον.

```

TEI xml output:—

```

<sp>
  <speaker>Σωκράτης</speaker>
  <p>κατέβην χθές εἰς Πειραιᾶ μετὰ Γλαύκωνος τοῦ Ἀρίστωνος
  προσευξόμενός τε τῇ θεῷ καὶ ἅμα τὴν ἑορτὴν βουλόμενος
  θεάσασθαι τίνα τρόπον ποιήσουσιν ἄτε νῦν πρῶτον ἄγοντες.
  καλὴ μὲν οὖν μοι καὶ ἡ τῶν ἐπιχωρίων πομπὴ ἔδοξεν εἶναι, οὐ
  μέντοι ἦττον ἐφαίνετο πρέπειν ἢν οἱ θοῤᾳκες ἔπεμπον.</p>
</sp>

```

## 13.7 References to Cited Works

A full example of what is technically called a *Conspectus Siglorum* can be found above in [sect. 2.4.1 on page 12](#). Such a list of manuscript sigla should be found immediately before the edition text. Traditionally, this section is followed by a list of other sources used to establish the text, so that the edited text is in the end established both from manuscript evidence (the “witnesses”) and other works based on a scholarly approach of the text (the “sources”) which are called in Latin *Editiones uel Studia*. As a consequence of this classification as “witness” or “source”, the former must go within the <listWit> element of the TEI header, whereas the latter is to be found within the <listBibl> element.

`\AddxmlBibResource` `\AddxmlBibResource{(basename or name.xml)}` is a preamble-only command. If a base name (either suffixed with `.xml` or not) for a TEI xml-compliant bibliographical database be provided, ekdosis will use it and insert formatted data in the back matter section of its own TEI xml output file, as <biblStruct> elements within a listBibl section.

As an example, the following Bib(L<sup>A</sup>)T<sub>E</sub>X entry and its TEI equivalent are provided:<sup>87</sup>—

<sup>87</sup> To the author’s knowledge, Zotero (<https://www.zotero.org>) provides excellent TEI xml output from Bib(L<sup>A</sup>)T<sub>E</sub>X input files.

```

1 @Book{Drak,
2   title = {Punicorum Libri Septemdecim},
3   author = {Silius Italicus, Tiberius Catius},
4   editor = {Drakenborch, Arnold},
5   date = {1717},
6   publisher = {Trajecti ad Rhenum},
7   location = {Utrecht}
8 }

```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <listBibl xmlns="http://www.tei-c.org/ns/1.0">
3   <biblStruct type="book" xml:id="Drak">
4     <monogr>
5       <title level="m">Punicorum libri septemdecim</title>
6       <author>
7         <forename>Tiberius Catius</forename>
8         <surname>Silius Italicus</surname>
9       </author>
10      <editor>
11        <forename>Arnold</forename>
12        <surname>Drakenborch</surname>
13      </editor>
14      <imprint>
15        <pubPlace>Utrecht</pubPlace>
16        <publisher>Trajecti ad Rhenum</publisher>
17        <date>1717</date>
18      </imprint>
19    </monogr>
20  </biblStruct>
21 </listBibl>

```

As can be seen, the same string `Drak` is used as a label in the Bib(L)TeX file (l. 1) and an `xml:id` in the TEI file (l. 3). This same label must be used again in the preamble of the `.tex` source file to declare Arnold Drakenborch as a source,<sup>88</sup> like so:—

```

1 % Use 'bibl.xml' as a TEI xml bibliographical database:
2 \AddxmlBibResource{bibdata.xml}
3
4 % Declare A. Drakenborch as source:
5 \DeclareSource{Drak}{\emph{Drakenborch}}

```

Finally, an extract of Silius Italicus' *Punica*, Book 9, ll. 30-2 follows (`.tex` source file, PDF output and TEI output files):—

```

1 % Preamble:
2 \usepackage[style=oxnotes]{biblatex}
3 \addbibresource{bibdata.bib}
4
5 \usepackage[telexport=tidy]{ekdosis}
6
7 % basename of the .xml bibliographical database:
8 \AddxmlBibResource{bibdata.xml}

```

88. See above on page 10.



```

9
10 % Witnesses:
11 \DeclareWitness{L}{L}{Laurentianus, plut, XXXVII, cod. 16}[
12     origDate=s. XV]
13 % Other witnesses [...]
14
15 % Sources:
16 \DeclareSource{Drak}{\emph{Drakenborch}}
17 % Alternatively, use BibLaTeX for the rendition:
18 % \DeclareSource{Drak}{\citename{Drak}{editor}}
19 % Other sources [...]
20
21 % Document:
22 \begin{ekdosis}
23   \begin{ekdverse}
24     Sed uos, quorum oculos atque ora humentia uidi,\!
25     uertere cum consul terga et remeare iuberet,\!
26     \app{
27       \lem[source=Drak, type=emendation]{ne morem}
28       \rdg[wit={L, F}]{me morem}
29       \rdg[wit={O, V}]{memorem}
30     } et pugnae signum exspectate petendae:\!
31   \end{ekdverse}
32 \end{ekdosis}

```

PDF output:—

Sed uos, quorum oculos atque ora humentia uidi,	30
uertere cum consul terga et remeare iuberet,	31
ne morem et pugnae signum exspectate petendae:	32

---

32 ne morem *Drakenborch*] me morem L F memorem O V

TEI xml output file produced by ekdosis (narrowed down to the <text> element):—

```

1 <text>
2   <body>
3     <lg>
4       <l>Sed uos, quorum oculos atque ora humentia uidi,</l>
5       <l>uertere cum consul terga et remeare iuberet,</l>
6       <l>
7         <app>
8           <lem source="#Drak" type="emendation">ne morem</lem>
9           <rdg wit="#L #F">me morem</rdg>
10          <rdg wit="#O #V">memorem</rdg>
11          </app>et pugnae signum exspectate petendae:</l>
12        </lg>
13      </body>
14    <back>
15      <listBibl>
16        <biblStruct type="book" xml:id="Drak">
17          <monogr>
18            <title level="m">Punicorum libri septemdecim</title>

```

```

19     <author>
20         <forename>Tiberius Catus</forename>
21         <surname>Silius Italicus</surname>
22     </author>
23     <editor>
24         <forename>Arnold</forename>
25         <surname>Drakenborch</surname>
26     </editor>
27     <imprint>
28         <pubPlace>Utrecht</pubPlace>
29         <publisher>Trajecti ad Rhenum</publisher>
30         <date>1717</date>
31     </imprint>
32 </monogr>
33 </biblStruct>
34 </listBibl>
35 </back>
36 </text>

```

## 13.8 Citation Commands

ekdosis can also convert into TEI `xml` references to cited works. Depending on the optional arguments used in the citation command, references will be converted into `<ptr>` or `<bibl>` elements with the appropriate identifier supplied by means of the `target` or `corresp` attributes.

Of course, for this mechanism to work, Bib<sub>TEX</sub> or Bib<sub>L</sub>TEX must be used and connected to some `.bib` bibliographical database file. Additionally, this `.bib` file must have been converted into a TEI `xml`-compliant file where all the Bib(L)TEX entries that are used in the document are found within `<biblStruct>` elements.<sup>89</sup> Finally, this `.xml` bibliographical database must have been connected to the `.tex` source file by means of `\AddxmlBibResource` described above in [sect. 13.7 on page 76](#).

As an example, the following `sample.bib` file is used:—

```

@Book{ReynoldsWilson1991,
  author = {Reynolds, L. D. and Wilson, N. G.},
  title = {Scribes and Scholars},
  year = {1991},
  subtitle = {A Guide to the Translation of Greek and Latin
             Literature},
  edition = {3},
  publisher = {Clarendon Press},
  location = {Oxford}
}

```

It has been converted into `sample.xml` as follows:—

```

<?xml version="1.0" encoding="UTF-8"?>
<listBibl xmlns="http://www.tei-c.org/ns/1.0">
  <biblStruct type="book" xml:id="ReynoldsWilson1991">
    <monogr>
      <title level="m">Scribes and Scholars</title>
      <author>

```

89. See above n. 87 on page 76 for information on how to do this.

```

    <forename>L. D.</forename>
    <surname>Reynolds</surname>
  </author>
  <author>
    <forename>N. G.</forename>
    <surname>Wilson</surname>
  </author>
  <edition>3</edition>
  <imprint>
    <pubPlace>Oxford</pubPlace>
    <publisher>Clarendon Press</publisher>
    <date>1991</date>
  </imprint>
</monogr>
</biblStruct>
</listBibl>

```

Once both files have been prepared, inserting references and exporting them into the TEI xml output file can be achieved in a straightforward way. (The full `sample.tex` is provided below.)—

```

\documentclass{article}

\usepackage[telexport=tidy]{ekdosis}
\AddxmlBibResource{sample.xml}

\usepackage[style=oxnotes]{biblatex}
\addbibresource{sample.bib}

\begin{document}
\begin{ekdosis}
  On textual criticism, see \cite[207--241]{ReynoldsWilson1991}.
\end{ekdosis}
\end{document}

```

PDF output:—

- 1 On textual criticism, see L. D. Reynolds and N. G. Wilson, *Scribes and Scholars: A*
- 2 *Guide to the Translation of Greek and Latin Literature* (3rd edn., Oxford: Clarendon Press,
- 3 1991), 207–41.

TEI xml output narrowed down to the contents of the `<text>` element:—

```

<text>
  <body>
    <p>On textual criticism, see
    <bibl corresp="#ReynoldsWilson1991">
      <biblScope>207--241</biblScope>
    </bibl>.</p>
  </body>
  <back>
    <listBibl>
      <biblStruct type="book" xml:id="ReynoldsWilson1991">
        <monogr>
          <title level="m">Scribes and Scholars</title>
          <author>

```

```

    <forename>L. D.</forename>
    <surname>Reynolds</surname>
  </author>
  <author>
    <forename>N. G.</forename>
    <surname>Wilson</surname>
  </author>
  <edition>3</edition>
  <imprint>
    <pubPlace>Oxford</pubPlace>
    <publisher>Clarendon Press</publisher>
    <date>1991</date>
  </imprint>
</monogr>
</biblStruct>
</listBibl>
</back>
</text>

```

At the time of writing, the following citation commands are converted into TEI `<xm1>` by ekdosis:—

- (a) `\icite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`<sup>90</sup>
- (b) `\cite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (c) `\Cite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (d) `\cite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (e) `\parencite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (f) `\Parencite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (g) `\parencite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (h) `\footcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (i) `\footcitetext[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (j) `\textcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (k) `\Textcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (l) `\smartcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (m) `\Smartcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (n) `\autocite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (o) `\Autocite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (p) `\autocite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`
- (q) `\Autocite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`

The next release of ekdosis will include all citation commands with the exception of so-called “qualified citation lists”.

## 14 Future Work

A short, un-commented list of what is planned in the versions of ekdosis to come follows:—

- (a) Very short-term (weeks):—
  - (a) Text structure: milestone elements.
  - (b) Marginal edition texts: It may happen that the marginalia of manuscripts contain texts worth editing in addition to and along the main text to which they are linked by reference signs.

---

90. From the icite package. `\icite` can be used in place of almost any standard citation command. See Alessi, *The Icite package* (cf. n. 79).

- (b) Short-term (months):—
  - (a) Poetry: `ekdosis` is now able to load and use the facilities provided by the `verse` package. Refined options will be added, such as metrical analysis. Arabic poetry through the environments and commands provided by the `arabluatex` package will also be supported. Other packages will also be considered for inclusion, such as `poetry` or `teubner`. In the end, `ekdosis` will provide a way for the typesetting of poetry which will allow for more flexibility and compatibility with TEI `xml`.
  - (b) Correspondence and alignment, segmentation: The functions are being tested at the time of writing and will be included in `ekdosis`.
- (c) Medium-term: Indexing, commands and environments for specific modules of the TEI.

## 15 GNU Free Documentation License

Version 1.3, 3 November 2008

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be

included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.
- H. Include an unaltered copy of this License.
  - I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
  - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.



You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

### **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.


If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

## 16 References

### Texts Used

- Caesar, *Gallic War*, ed. W. A. McDevitte and W. S. Bohn (Harper’s New Classical Library; 1st edn., New York: Harper & Brothers, 1869).
- *Gallic War (Guerre des Gaules)*, ed. L.-A. Constans (Collection des Universités de France; Paris: Les Belles Lettres, 1987) (originally pub. 1926).
- Cicero, *Ad Atticum epistularum libri sedecim*, recensuit H. Sjögren (Collectio scriptorum ueterum Vpsaliensis; Eranos’ Förlag, 1916).
- *Letters to Atticus*, ed. E.O. Winstedt (The Loeb Classical Library, 2; London – New York: William Heinemann & The MacMillan Co., 1919).
- Homer, *The Odyssey*, ed. A. T. Murray, 2 vols. (Cambridge, MA. – London: Harvard University Press – William Heinemann, 1919).
- Shakespeare, *The Sonnets* [From the Quarto of 1609 with Variorum Readings and Commentary], ed. Raymond MacDonald Alden (Boston & New York: The Riverside Press Cambridge, 1916).

### References

- Alessi, Robert, “ekdosis: Using LuaL<sup>A</sup>T<sub>E</sub>X for Producing TEI xml-Compliant Critical Editions and Highlighting Parallel Writings,” *Journal of Data Mining and Digital Humanities: Collecting, Preserving, and Disseminating Endangered Cultural Heritage for New Understandings through Multilingual Approaches* (Nov. 2020), [jdmhd: 6536](#). 
- *Old Standard: A Unicode Font for Classical and Medieval Studies* (version 2.6) [Based on Alexey Kryukov’s original Old Standard] (Dec. 18, 2020), <http://ctan.org/pkg/oldstandard>.
- *The ArabluateX package* (version 1.20) [ArabTeX for LuaLaTeX] (Mar. 23, 2020), <http://ctan.org/pkg/arabluateX>.
- *The Icite package* (version 1.3a) [Indices locorum citatorum] (Mar. 5, 2020), <http://ctan.org/pkg/icite>.
- Bezoz López, Javier, *The Titleps package* (version 2.13) [Page style control] (Oct. 16, 2019), <https://ctan.org/pkg/titleps>.
- Bezoz López, Javier and Braams, Johannes L., *The Babel package* (version 3.47) [Multilingual support for Plain TeX or LaTeX] (July 13, 2020), <http://www.ctan.org/pkg/babel>.
- Charette, François and Reutenauer, Arthur, *The Polyglossia package* (version 1.49) [An alternative to babel for XeLaTeX and LuaLaTeX] (Apr. 8, 2020), <http://www.ctan.org/pkg/polyglossia>.
- Digital Latin Library, “Textual Criticism,” <https://digitallatin.org/library-digital-latin-texts/textual-criticism>, accessed May 24, 2020.
- Dunn, Brian, *The Keyfloat package* (version 2.06) [Provides a key/value interface for generating floats] (June 29, 2021), <https://ctan.org/pkg/keyfloat>.
- Kohm, Markus, *The Marginnote package* (version 1.4b) [Notes in the margin, even where \marginpar fails] (Aug. 9, 2018), <https://komascript.de/marginnote>.

- Lagally, Klaus, *The Arabtex package* (version 4.00) [Macros and fonts for typesetting Arabic] (Mar. 11, 2004), [http://baobab.informatik.uni-stuttgart.de/ifi/bs/research/arab\\_e.html](http://baobab.informatik.uni-stuttgart.de/ifi/bs/research/arab_e.html).
- Lück, Uwe and Böttcher, Stephan, *The Lineno package* (version 4.41) [Line numbers on paragraphs] (Nov. 2, 2005), <http://www.ctan.org/pkg/lineno>.
- Madsen, Lars and Wilson, Peter R., *The Memoir package* (version 3.7o) [Typeset fiction, non-fiction and mathematical books] (Mar. 23, 2021), <http://www.ctan.org/pkg/memoir>.
- Mastronarde, Donald J. and Tarrant, Richard J., “Review: Guidelines for Encoding Critical Editions for the Library of Digital Latin Texts,” *Society for Classical Studies* (Dec. 4, 2017), <https://classicalstudies.org/scs-blog/donald-j-mastronarde/review-guidelines-encoding-critical-editions-library-digital-latin>.
- Nakashima, Hiroshi, *The Fnpos package* (version 1.0) [Control the position of footnotes on the page] (Sept. 3, 2018), <http://www.ctan.org/pkg/fnpos>.
- *The Paracol package* (version 1.35) [Multiple columns with texts “in parallel”] (Dec. 31, 2018), <http://www.ctan.org/pkg/paracol>.
- Reynolds, L. D. and Wilson, N. G., *Scribes and Scholars: A Guide to the Translation of Greek and Latin Literature* (3rd edn., Oxford: Clarendon Press, 1991).
- Rozhenko, Alexander I., *The Extdash package* (version 1.3) [A range of dash commands for compound words] (June 24, 2018), <http://www.ctan.org/pkg/extdash>.
- Sturm, Thomas F., *The Tcolorbox package* (version 4.51) [Coloured boxes, for LaTeX examples and theorems, etc] (June 14, 2021), <https://ctan.org/pkg/tcolorbox>.
- The LaTeX Team, *The Xparse package* [A generic document command parser] (Jan. 12, 2022), <https://ctan.org/pkg/xparse>.
- Van den Branden, Ron, Terras, Melissa, and Vanhoutte, Edward, “TEI by Example,” <http://www.teibyexample.org>, accessed Aug. 4, 2020.
- West, Martin L., *Textual Criticism and Editorial Technique* [Applicable to Greek and Latin Texts] (Stuttgart: B. G. Teubner, 1973).
- Wilson, Peter R. and Robertson, Will, *The Verse package* (version 2.4b) [Aids for typesetting simple verse] (May 10, 2014), <http://www.ctan.org/pkg/verse>.

## 17 Sample: C. J. Caesar, *Gallic War*, VI, XIII.1

### 17.1 .tex Source File

```

1 \documentclass[12pt]{article}
2 \usepackage{fontspec}
3 \usepackage[latin.classic,french,english]{babel}
4 \babelfont{rm}{Old Standard}
5 \babelfont{sf}{NewComputerModernSans10}
6 \babelfont{tt}{NewComputerModernMono10}
7
8 \usepackage{nextpage}
9 \usepackage{xltabular}
10
11 \usepackage[telexport=tidy]{ekdosis}
12 \DeclareApparatus{default}[

```

```

13     delim=\hskip0.75em,
14     ehook=.]
15
16 \FormatDiv{2}{.}
17
18 \SetAlignment{
19     tcols=3,
20     lcols=1,
21     texts=latin[xml:lang="la"];
22     english[xml:lang="en"];
23     french[xml:lang="fr"],
24     apparatus=latin,
25     segmentation=auto}
26
27 \AtBeginEnvironment{latin}{\selectlanguage{latin}}
28 \AtBeginEnvironment{english}{\sloppy\selectlanguage{english}}
29 \AtBeginEnvironment{french}{\sloppy\selectlanguage{french}}
30
31 \DeclareWitness{A}{A}{\emph{Bongarsianus} 81}[
32     msName=Bongarsianus,
33     settlement=Amsterdam,
34     idno=81,
35     institution=University Library,
36     origDate=s. IX--X]
37 \DeclareHand{A1}{A}{A\textsuperscript{1}}[\emph{Emendationes}
38     scribae ipsius]
39 \DeclareWitness{M}{M}{\emph{Parisinus Lat.} 5056}[
40     origDate={s. XII}]
41 \DeclareWitness{B}{B}{\emph{Parisinus Lat.} 5763}[
42     origDate={s. IX--X}]
43 \DeclareWitness{R}{R}{\emph{Vaticanus Lat.} 3864}[
44     origDate={s. X}]
45 \DeclareWitness{S}{S}{\emph{Laurentianus} R 33}[
46     origDate={s. X}]
47 \DeclareWitness{L}{L}{\emph{Londinensis} Br. Mus. 10084}[
48     origDate={s. XI}]
49 \DeclareWitness{N}{N}{\emph{Neapolitanus} IV, c. 11}[
50     origDate={s. XII}]
51 \DeclareWitness{T}{T}{\emph{Parisinus Lat.} 5764}[
52     origDate={s. XI}]
53 \DeclareWitness{f}{f}{\emph{f}}{\emph{Vindobonensis} 95}[
54     origDate={s. XII}]
55 \DeclareWitness{U}{U}{\emph{Vaticanus Lat.} 3324}[
56     origDate={s. XI}]
57 \DeclareWitness{l}{l}{\emph{l}}{\emph{Laurentianus} Riccard. 541}[
58     origDate={s. XI--XII}]
59 \DeclareShorthand{a}{\alpha}{A,M,B,R,S,L,N}
60 \DeclareShorthand{b}{\beta}{T,f,U,l}
61
62 \begin{document}
63
64 \begin{xltabular}[c]{0.75\linewidth}{lXl}
65     \caption*{\textbf{Conspectus siglorum}\label{tab:conspectus-siglorum}}\
66     \multicolumn{3}{c}{\emph{Familia} \getsiglum{a}}\
67     \SigLine{A}\

```

```

68 & \getsiglum{A1} \emph{Emendationes scribae ipsius} & \\
69 \SigLine{M}\\
70 \SigLine{B}\\
71 \SigLine{R}\\
72 \SigLine{S}\\
73 \SigLine{L}\\
74 \SigLine{N}\\
75 \multicolumn{3}{c}{\emph{Familia} \getsiglum{b}}\\
76 \SigLine{T}\\
77 \SigLine{f}\\
78 \SigLine{U}\\
79 \SigLine{l}\\
80 \end{xltabular}
81
82 \cleartoevenpage
83
84 \begin{alignment}
85 \begin{latin}
86 \ekddiv{head=XIII, depth=2, n=6.13, type=section}
87 In omni Gallia eorum hominum qui \app{
88 \lem[wit=a]{aliquo}
89 \rdg[wit=b, alt=in al-]{in aliquo}}
90 sunt numero atque honore genera sunt duo. Nam plebes paene
91 seruorum habetur loco, quae \app{
92 \lem[wit={A,M}, alt={nihil audet (aut et \getsiglum{A1})}
93 per se]{nihil audet per se}
94 \rdg[wit=A1,nordg]{nihil aut et per se}
95 \rdg[wit={R,S,L,N}]{nihil habet per se}
96 \rdg[wit=b]{per se nihil audet}}, \app{
97 \lem[wit=a]{nullo}
98 \rdg[wit=b]{nulli}} adhibetur \app{
99 \lem{consilio}
100 \rdg[wit={T, U}, alt=conc-]{concilio}}.
101 \end{latin}
102 \begin{english}
103 \ekddiv{head=XIII, depth=2, n=6.13, type=section}
104 Throughout all Gaul there are two orders of those men who are of
105 any rank and dignity: for the commonality is held almost in the
106 condition of slaves, and dares to undertake nothing of itself,
107 and is admitted to no deliberation.
108 \end{english}
109 \begin{french}
110 \ekddiv{head=XIII, depth=2, n=6.13, type=section}
111 Partout en Gaule il y a deux classes d'hommes qui comptent et qui
112 sont considérés. Quant aux gens du peuple, ils ne sont guère
113 traités autrement que des esclaves, ne pouvant se permettre aucune
114 initiative, n'étant consultés sur rien.
115 \end{french}
116 \end{alignment}
117
118 \end{document}

```

## 17.2 TEI xml Output

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <TEI xmlns="http://www.tei-c.org/ns/1.0">
3   <teiHeader>
4     <fileDesc>
5       <titleStmt>
6         <title>
7           <!-- Title -->
8         </title>
9       <respStmt>
10        <resp>
11          <!-- Edited by -->
12        </resp>
13        <name>
14          <!-- Name -->
15        </name>
16      </respStmt>
17    </titleStmt>
18    <publicationStmt>
19      <distributor>
20        <!-- Distributor name -->
21      </distributor>
22    </publicationStmt>
23    <sourceDesc>
24      <listWit>
25        <witness xml:id="A">
26          <abbr type="siglum">A</abbr>
27          <emph>Bongarsianus</emph>81
28        <msDesc>
29          <msIdentifier>
30            <settlement>Amsterdam</settlement>
31            <institution>University Library</institution>
32            <idno>81</idno>
33            <msName>Bongarsianus</msName>
34          </msIdentifier>
35          <physDesc>
36            <handDesc hands="1">
37              <handNote xml:id="A1">
38                <abbr type="siglum">A
39                <hi rend="sup">1</hi></abbr>
40                <p>
41                  <emph>Emendationes scribae ipsius</emph>
42                </p>
43              </handNote>
44            </handDesc>
45          </physDesc>
46          <history>
47            <origin>
48              <origDate>s. IX--X</origDate>
49            </origin>
50          </history>
51        </msDesc></witness>
52        <witness xml:id="M">
53          <abbr type="siglum">M</abbr>
```



```

54      <emph>Parisinus Lat.</emph>5056
55      <msDesc>
56        <msIdentifier />
57        <history>
58          <origin>
59            <origDate>s. XII</origDate>
60          </origin>
61        </history>
62      </msDesc></witness>
63      <witness xml:id="B">
64        <abbr type="siglum">B</abbr>
65      <emph>Parisinus Lat.</emph>5763
66      <msDesc>
67        <msIdentifier />
68        <history>
69          <origin>
70            <origDate>s. IX--X</origDate>
71          </origin>
72        </history>
73      </msDesc></witness>
74      <witness xml:id="R">
75        <abbr type="siglum">R</abbr>
76      <emph>Vaticanus Lat.</emph>3864
77      <msDesc>
78        <msIdentifier />
79        <history>
80          <origin>
81            <origDate>s. X</origDate>
82          </origin>
83        </history>
84      </msDesc></witness>
85      <witness xml:id="S">
86        <abbr type="siglum">S</abbr>
87      <emph>Laurentianus</emph>R 33
88      <msDesc>
89        <msIdentifier />
90        <history>
91          <origin>
92            <origDate>s. X</origDate>
93          </origin>
94        </history>
95      </msDesc></witness>
96      <witness xml:id="L">
97        <abbr type="siglum">L</abbr>
98      <emph>Londinensis</emph>Br. Mus. 10084
99      <msDesc>
100       <msIdentifier />
101       <history>
102         <origin>
103           <origDate>s. XI</origDate>
104         </origin>
105       </history>
106     </msDesc></witness>
107     <witness xml:id="N">
108       <abbr type="siglum">N</abbr>

```

```

109 <emph>Neapolitanus</emph>IV, c. 11
110 <msDesc>
111 <msIdentifier />
112 <history>
113 <origin>
114 <origDate>s. XII</origDate>
115 </origin>
116 </history>
117 </msDesc></witness>
118 <witness xml:id="T">
119 <abbr type="siglum">T</abbr>
120 <emph>Parisinus Lat.</emph>5764
121 <msDesc>
122 <msIdentifier />
123 <history>
124 <origin>
125 <origDate>s. XI</origDate>
126 </origin>
127 </history>
128 </msDesc></witness>
129 <witness xml:id="f">
130 <abbr type="siglum">
131 <emph>f</emph>
132 </abbr>
133 <emph>Vindobonensis</emph>95
134 <msDesc>
135 <msIdentifier />
136 <history>
137 <origin>
138 <origDate>s. XII</origDate>
139 </origin>
140 </history>
141 </msDesc></witness>
142 <witness xml:id="U">
143 <abbr type="siglum">U</abbr>
144 <emph>Vaticanus Lat.</emph>3324
145 <msDesc>
146 <msIdentifier />
147 <history>
148 <origin>
149 <origDate>s. XI</origDate>
150 </origin>
151 </history>
152 </msDesc></witness>
153 <witness xml:id="l">
154 <abbr type="siglum">
155 <emph>l</emph>
156 </abbr>
157 <emph>Laurentianus</emph>Riccard. 541
158 <msDesc>
159 <msIdentifier />
160 <history>
161 <origin>
162 <origDate>s. XI--XII</origDate>
163 </origin>

```

```

164         </history>
165     </msDesc></witness>
166 </listWit>
167 </sourceDesc>
168 </fileDesc>
169 <encodingDesc>
170     <variantEncoding method="parallel-segmentation"
171         location="internal" />
172 </encodingDesc>
173 </teiHeader>
174 <text>
175     <body>
176         <div xml:id="div-latin_1" xml:lang="la">
177             <div type="section" n="6.13">
178                 <head>XIII</head>
179                 <p>In omni Gallia eorum hominum qui
180                 <app>
181                     <lem wit="#A #M #B #R #S #L #N">aliquo</lem>
182                     <rdg wit="#T #f #U #l">in aliquo</rdg>
183                 </app>sunt numero atque honore genera sunt duo. Nam
184                 plebes paene seruorum habetur loco, quae
185                 <app>
186                     <lem wit="#A #M">nihil audet per se</lem>
187                     <rdg wit="#A1">nihil aut et per se</rdg>
188                     <rdg wit="#R #S #L #N">nihil habet per se</rdg>
189                     <rdg wit="#T #f #U #l">per se nihil audet</rdg>
190                 </app>,
191                 <app>
192                     <lem wit="#A #M #B #R #S #L #N">nullo</lem>
193                     <rdg wit="#T #f #U #l">>nulli</rdg>
194                 </app>adhibetur
195                 <app>
196                     <lem>consilio</lem>
197                     <rdg wit="#T #U">concilio</rdg>
198                 </app>.</p>
199             </div>
200         </div>
201         <div xml:id="div-english_1" xml:lang="en">
202             <div type="section" n="6.13">
203                 <head>XIII</head>
204                 <p>Throughout all Gaul there are two orders of those men
205                 who are of any rank and dignity: for the commonality is
206                 held almost in the condition of slaves, and dares to
207                 undertake nothing of itself, and is admitted to no
208                 deliberation.</p>
209             </div>
210         </div>
211         <div xml:id="div-french_1" xml:lang="fr">
212             <div type="section" n="6.13">
213                 <head>XIII</head>
214                 <p>Partout en Gaule il y a deux classes d'hommes qui
215                 comptent et qui sont considérés. Quant aux gens du
216                 peuple, ils ne sont guère traités autrement que des
217                 esclaves, ne pouvant se permettre aucune initiative,
218                 n'étant consultés sur rien.</p>

```

```
219     </div>
220   </div>
221 </body>
222 </text>
223 </TEI>
```

## 18 Arabic Sample File

arabic-sample.tex:—

```
% Instructions:
% 1. Compile this file three times.
%   - Open arabic-sample.pdf and arabic-sample-tei.xml and see the
%     result.
% 2. Compile arabic-sample_out.tex three times.
%   - Open arabic-sample_out.pdf and arabic-sample-out-tei.xml and
%     see the result.
%
\documentclass{article}

% The following three lines are only needed by the
% 'arabic-sample_out.tex' that arabluatex is instructed to produce:
\usepackage{babel}
\babelprovide[onchar=fonts]{arabic}
\babelfont[*arabic]{rm}{Amiri}

% instruct ekdosis to output TEI xml (arabic-sample-tei.xml):
\usepackage[telexport=tidy]{ekdosis}

% instruct arabluatex to output sample-arabic_out.tex with Unicode
% Arabic strings in place of arabtex ASCII scheme:
\usepackage[fullvoc,export]{arabluatex}

\begin{document}

\begin{arabexport} % export arabtex strings to Unicode Arabic
  \begin{ekdosis}
    \begin{arab}
      'inna 'abI kAna mina
      \app{
        \lem{'l-muqAtilaTi}
        \rdg{'l-muqAtilIna}
      }
      wa-kAnat 'ummI min `u.zamA'i buyUti 'l-zamAzimaTi.
    \end{arab}
  \end{ekdosis}
\end{arabexport}
\end{document}
```

## 19 Implementation

ekdosis relies on Lua functions and tables. Read the .lua files that accompany ekdosis for more information.

```
1 \RequirePackage{iftex}
```

Of course, ekdosis requires Lua<sup>A</sup>T<sub>E</sub>X. Issue an error if the document is processed with another engine.

```
2 \RequireLuaTeX
```

Set global options:—

```
3 \RequirePackage{expkv-opt}
4 \RequirePackage{expkv-def}
5 \newif\if@pkg@float
6 \newif\if@pkg@footins
7 \newif\if@pkg@keyfloat
8 \newif\if@pkg@fitapp
9 \newif\if@pkg@breakable
10 \newif\if@pkg@ekddivs
11 \newif\if@parnotesroman
12 \newif\if@pkg@parnotes
13 \newif\iftei@export
14 \newif\if@pkg@poetry@verse
15 \ekvdefinekeys{ekdosis}{
16   choice layout = {float = {\@pkg@floattrue},
17     footins = {\@pkg@floatfalse\@pkg@footintrue},
18     keyfloat = {\@pkg@floatfalse\@pkg@keyfloattrue},
19     fitapp = {\@pkg@floatfalse\@pkg@fitapptrue},
20     breakable = {\@pkg@floatfalse\@pkg@fitapptrue\@pkg@breakabletrue}},
21   initial layout = float,
22   unknown-choice layout = \PackageError{ekdosis}{unknown
23     layout=#1}{`layout' must be either `float', `footins', `keyfloat',
24     `fitapp' or `breakable'.},
25   choice divs = {ekdosis = {\@pkg@ekddivstrue},
26     latex = {\@pkg@ekddivfalse
27       \AtBeginDocument{\luadirect{ekdosis.setekddivfalse()}}}},
28   initial divs = ekdosis,
29   unknown-choice divs = \PackageError{ekdosis}{unknown divs=#1}{`divs'
30     must be either `ekdosis' or `latex'.},
31   choice poetry = {verse = {\@pkg@poetry@versetrue}},
32   unknown-choice poetry = \PackageError{ekdosis}{unknown
33     poetry=#1}{`poetry' must be `verse' for now.},
34   choice parnotes = {false = {}},
35     true = {\@pkg@parnotestrue},
36     roman = {\@pkg@parnotestrue\@parnotesromantrue}},
37   default parnotes = true,
38   unknown-choice parnotes = \PackageError{ekdosis}{unknown
39     parnotes=#1}{`parnotes' must be either `true', or `false' or
40     `roman'.},
41   choice teiexport = {false = {}},
42     true = {\tei@exporttrue
43       \AtBeginDocument{\luadirect{ekdosis.openteistream()}}%
44       \AtEndDocument{\luadirect{ekdosis.closeiteistream()}}},
45   tidy = {\tei@exporttrue
46     \AtBeginDocument{\luadirect{ekdosis.openteistream()}}%
47     \AtEndDocument{\luadirect{ekdosis.closeiteistream("tidy")}}},
```

```

48 default teiexport = true,
49 unknown-choice teiexport = \PackageError{ekdosis}{unknown
50   teiexport=#1}{`teiexport' must be either `true', `false' or
51   `tidy'`}
52 }
53 \ekvoProcessLocalOptions{ekdosis}
54 \newif\ifekd@memoir@loaded
55 \@ifclassloaded{memoir}{%
56   \ekd@memoir@loadedtrue\@pkg@poetry@versetrue}{%

```

**Required Packages** In addition to iftex, expkv-opt and expkv-def, a list of the packages that are required by ekdosis follows:—

```

57 % \RequirePackage{iftex} % already loaded above
58 % \RequirePackage{expkv-opt} % already loaded above
59 % \RequirePackage{expkv-def} % already loaded above
60 \RequirePackage{luacode}
61 \RequirePackage{paracol}
62 \RequirePackage{etoolbox}
63 \RequirePackage{lineno}
64 \if@pkg@float
65   \RequirePackage{trivfloat}
66   \trivfloat{ekdapparatus}
67 \fi
68 \if@pkg@keyfloat
69   \RequirePackage{keyfloat}
70   \def\ekd@keyparopts#1{%
71     \def\ekd@insert@keyparapp{%
72       \keyparbox[!b]{#1}{\ekd@insert@apparatus}}
73   \ekd@keyparopts}
74 \fi
75 \if@pkg@fitapp
76   \RequirePackage{tcolorbox}
77   \tcbuselibrary{fitting,skins}
78 \fi
79 \if@pkg@breakable
80   \RequirePackage{tcolorbox}
81   \tcbuselibrary{fitting,skins,breakable}
82 \fi
83 \RequirePackage{refcount}
84 \RequirePackage{zref-user}
85 \RequirePackage{zref-abspage}
86 \RequirePackage{ltxcmds}
87 \RequirePackage{pdftexcmds}
88 \RequirePackage{ifoddpages}
89 \if@pkg@poetry@verse
90   \RequirePackage{verse}
91 \fi
92 \if@pkg@parnotes
93   \RequirePackage{parnotes}
94 \fi

```

**Lua** Here begins the real work: load ekdosis.lua:—

```

95 \luadirect{dofile(kpse.find_file("ekdosis.lua"))}
96 \AtEndDocument{

```

```

97 \luadirect{ekdosis.closestream()}
98 }

```

### ekdosis Symbol

`\eKd` As of v1.5, `ekdosis` has its own identifying symbol. It is produced by `\eKd` and best printed with the Old Standard Greek font.

```

99 \def\eKd{%
100  ε\kern -.4em\raise 1.15ex\hbox{κ}\kern -.105emδ%
101  \ifdefined\xspace\xspace\fi
102 }

```

### Setup

`\ekdsetup` `\ekdsetup` is used to specify options that affect the general behavior of `ekdosis`. It is a preamble-only command.

```

103 \ekvdefinekeys{ekd@setup}{
104   bool showpagebreaks = \ifekd@showpb,
105   store spbm = \ekd@spbm,
106   initial spbm = spb,
107   store hpbm = \ekd@hpbm,
108   initial hpbm = hpb,
109 }
110 \NewDocumentCommand{\ekdsetup}{m}{\ekvset{ekd@setup}{#1}}
111 \@onlypreamble\ekdsetup

```

`\SetHooks` `\SetHooks` is used to set hooks meant to be shared by all declared apparatuses, such as the font size, the format of numerals, &c. This command can be used in the preamble or at any point of the document.

```

112 \ekvdefinekeys{ekd@hooks}{
113   store appfontsize = \ekd@appfontsize,
114   store refnumstyle = \ekd@refnumstyle,
115   store postrefnum = \ekd@postrefnum,
116   code familysep = \luadirect{ekdosis.setfamilysep(\luastringN{#1})},
117   store lemmastyle = \ekd@lemmastyle,
118   store readingstyle = \ekd@readingstyle,
119   code initialrule = \def\ekd@initial@rule{#1\NLS},
120   default initialrule = \rule{0.4\columnwidth}{0.4pt},
121   noval noinitialrule = \undef\ekd@initial@rule,
122   code maxentries = \luadirect{ekdosis.setglimit(\luastringN{#1})},
123   mmeta nomaxentries = {maxentries=none},
124   code keyparopts = \if@pkg@keyfloat\ekd@keyparopts{#1}\fi,
125   dimen appheight = \ekd@app@height,
126   initial appheight = .5\textheight,
127   choice fitalgorithm = {fontsize = \def\ekd@fit@algorithm{fontsize},
128     hybrid = \def\ekd@fit@algorithm{hybrid},
129     areasize = \def\ekd@fit@algorithm{areasize},
130     squeeze = \def\ekd@fit@algorithm{squeeze}},
131   initial fitalgorithm = fontsize,
132   unknown-choice fitalgorithm = \PackageError{ekdosis}{unknown
133     fitalgorithm=#1}{`fitalgorithm' must be either `fontsize',
134     `hybrid', `areasize' or `squeeze'.},
135   initial appfontsize = \footnotesize,
136   initial refnumstyle = \bfseries,

```

```

137 initial postrefnum = ~,
138 initial lemmastyle = {},
139 initial readingstyle = {}
140 }
141 \NewDocumentCommand{\SetHooks}{m}{\ekvset{ekd@hooks}{#1}}

```

Build and process the list of witnesses and hands:—

```

142 \ekvdefinekeys{ekd@witness}{
143   store settlement = \settlement@value,
144   store institution = \institution@value,
145   store repository = \repository@value,
146   store collection = \collection@value,
147   store idno = \idno@value,
148   store msName = \msName@value,
149   store origDate = \origDate@value,
150   store locus = \locus@value
151 }

```

`\DeclareWitness` `\DeclareWitness` is a preamble-only command. It takes three mandatory arguments and one optional argument. It is meant to collect data related to witnesses to be used in the edition text. Data are stored in Lua tables and are used to encode the `<listWit>` part of the TEI header as well as the *Conspectus Siglorum* in the edition in print.

```

152 \NewDocumentCommand{\DeclareWitness}{m m m O{}}{%
153   \bgroup
154   \ekvset{ekd@witness}{#4}
155   \luadirect{ekdosis.newwitness(
156     \luastringN{#1},
157     \luastringN{#2},
158     \luastringN{#3},
159     \luastringO{\settlement@value},
160     \luastringO{\institution@value},
161     \luastringO{\repository@value},
162     \luastringO{\collection@value},
163     \luastringO{\idno@value},
164     \luastringO{\msName@value},
165     \luastringO{\origDate@value},
166     \luastringO{\locus@value})}
167   \egroup
168   }
169 \@onlypreamble\DeclareWitness

```

`\DeclareHand` As `\DeclareWitness`, `\DeclareHand` is a preamble-only command meant to collect data and store them in Lua tables. It takes three mandatory arguments and one optional argument. The second argument is used to connect the hand to a declared witness it is related to. Then the table in which this witness is recorded can be fed with new data.

```

170 \NewDocumentCommand{\DeclareHand}{m m m +O{}}{
171   \luadirect{ekdosis.newhand(\luastringN{#1},
172     \luastringN{#2},
173     \luastringN{#3},
174     \luastringN{#4})}
175 }
176 \@onlypreamble\DeclareHand

```

Build and process the list of scholars:—

```

177 \ekvdefinekeys{ekd@scholar}{

```



```

178 store rawname = \rawname@value,
179 store forename = \forename@value,
180 store surname = \surname@value,
181 store addname = \addname@value,
182 store note = \note@value
183 }

```

`\DeclareScholar` `\DeclareScholar` is used to build a list of persons within the `<listPerson>` element. It takes two mandatory arguments to specify consecutively a unique identifier and the rendition to be used in the apparatus criticus in print, and one optional argument used to collect the name parts components and further items of information from key-value ‘named’ arguments.

```

184 \NewDocumentCommand{\DeclareScholar}{m m O{}}{%
185   \bgroup
186   \ekvset{ekd@scholar}{#3}
187   \luadirect{ekdosis.newscholar(
188     \luastringN{#1},
189     \luastringN{#2},
190     \luastringO{\rawname@value},
191     \luastringO{\forename@value},
192     \luastringO{\surname@value},
193     \luastringO{\addname@value},
194     \luastringO{\note@value})}
195   \egroup
196 }
197 \@onlypreamble\DeclareScholar

```

`\DeclareSource` There is also a table in which are collected data related to sources to be used in the apparatus criticus. `\DeclareSource` is a preamble-only command and takes two mandatory arguments: a unique id and a shorthand (preferably a Bib(L<sup>A</sup>T<sub>E</sub>X label) to be used in the apparatus criticus which can be extracted from a bibliographic database.

```

198 \NewDocumentCommand{\DeclareSource}{m m}{
199   \luadirect{ekdosis.newsource(\luastringN{#1},
200     \luastringN{#2})}
201 }
202 \@onlypreamble\DeclareSource

```

`\DeclareShorthand` `\DeclareShorthand` is a preamble-only command that can be used to record manuscript families or any kind of shorthand to be used to refer to previously declared ids, for example the shorthand `codd` can be used to point to all declared witnesses. This command takes three mandatory arguments: a unique id, its rendition in print and a csv-list of previously declared ids.

```

203 \NewDocumentCommand{\DeclareShorthand}{m m m}{
204   \luadirect{ekdosis.newshorthand(\luastringN{#1},
205     \luastringN{#2},
206     \luastringN{#3})}
207 }
208 \@onlypreamble\DeclareShorthand

```

`\getsiglum` `\getsiglum{<csv list>}` takes a comma-separated list of declared ids by means of `\DeclareWitness`, `\DeclareHand`, `\DeclareShorthand` or `\DeclareSource` and returns their respective renditions.

```

209 \NewDocumentCommand{\getsiglum}{m}{%
210   \luadirect{tex.sprint(ekdosis.getsiglum(\luastringN{#1}))}%
211 }

```

`\SigLine` `\Sigline`{*unique id*} takes the unique id of any declared witness by means of `\DeclareWitness` as argument and returns a line ready to be inserted in a table set to print a Conspectus Siglorum. `\SigLine` returns three fields separated by the symbol & that is used in tables as follows: the siglum referring to the witness, the contents of the description field and the contents of the optional `origDate` field.

```
212 \NewDocumentCommand{\SigLine}{m}{%
213   \luadirect{tex.sprint(ekdosis.basic_cs(\luastringN{#1}))}
214 }
```

**TEX to TEI xml** Here follow the key-value options to be used by `\SetTEIxmlExport` below:—

```
215 \ekvdefinekeys{tei@settings}{
216   choice autopar = {true = \luadirect{ekdosis.setteiautopar("yes")},
217     false = {\luadirect{ekdosis.setteiautopar("no")}}},
218   initial autopar = true,
219   unknown-choice autopar = \PackageError{ekdosis}{unknown
220     autopar=#1}{`autopar' must be either `true' or `false'.}
221 }
```

`\SetTEIxmlExport` `\SetTEIxmlExport` collects the settings to be applied to TEI xml export. For now, there is only one option. This command can be used at any point of the document, except inside environments meant to receive an apparatus criticus.

```
222 \NewDocumentCommand{\SetTEIxmlExport}{m}{
223   \unless\ifekd@state\ekvset{tei@settings}{#1}\fi
224 }
```

The following three commands can be used to instruct `ekdosis` how to convert unknown or unusual  $\LaTeX$  commands into TEI xml equivalents.

`\TeXtoTEI` `\TeXtoTEI`{*csname*}{*TEI element*}[*TEI attribute(s)*] takes two mandatory arguments and one optional argument, namely: the control sequence name to be converted, the TEI element it is to be converted into and any additional xml attributes to be appended to the opening TEI element:—

```
225 \NewDocumentCommand{\TeXtoTEI}{m m O{}}{%
226   \luadirect{ekdosis.newcmdtotag(\luastringN{#1},
227     \luastringN{#2},
228     \luastringN{#3})}
229 }
```

`\teidirect` `\teidirect`[*xml attributes*]{*xml element*}{*code*} does nothing in  $\LaTeX$ . It is only used to insert elements in the TEI xml output file. `\teidirectE`[*xml attributes*]{*xml element*} is strictly equivalent to `\teidirect`[*xml attributes*]{*xml element*}{} and can be used to insert empty TEI elements.

```
230 \NewDocumentCommand{\teidirect}{0{ } m m}{\ignorespaces}
231 \NewDocumentCommand{\teidirectE}{0{ } m}{\ignorespaces}
```

`\getTEIxmlid` This command returns from a csv-list of unique identifiers declared in commands such as `\DeclareWitness` and the like a space-separated list of their corresponding xml:ids, each preceded by the octothorpe (the # sign).

```
232 \NewDocumentCommand{\getTEIxmlid}{m}{%
233   \luadirect{tex.sprint(ekdosis.getsiglum(\luastringN{#1}, "TEI"))}%
234 }
```

`\EnvtoTEI` `\EnvtoTEI*` $\langle env name \rangle$  $\langle TEI element \rangle$  $[\langle TEI attribute(s) \rangle]$  instructs how to convert  $\LaTeX$  environments into TEI `xml` equivalents. It takes two mandatory arguments and one optional argument, namely the name of the  $\LaTeX$  environment to be converted, the TEI element it is to be converted into and any additional attributes to be appended to the TEI opening element. `\EnvtoTEI*` is restricted to TEI elements that must never appear within `<p>` elements, such as `<div>`, `<lg>` and the like.

```

235 \NewDocumentCommand{\EnvtoTEI}{s m m O{}}{%
236   \IfBooleanTF{#1}{%
237     \luadirect{ekdosis.newenvtotag(\luastringN{#2},
238       \luastringN{#3},
239       \luastringN{#4},
240       "yes")}
241   }{%
242     \luadirect{ekdosis.newenvtotag(\luastringN{#2},
243       \luastringN{#3},
244       \luastringN{#4})}
245   }
246 }

```

`\TeXtoTEIPat` Finally, the more flexible—and more delicate to handle—`\TeXtoTEIPat` $\langle TEX pattern \rangle$  $\langle TEI pattern \rangle$  uses pattern matching to instruct `ekdosis` how to convert  $\LaTeX$  commands into TEI equivalents.

```

247 \NewDocumentCommand{\TeXtoTEIPat}{m m}{%
248   \luadirect{ekdosis.newpatttotag(\luastringN{#1}, \luastringN{#2})}
249 }

```

`\SetTEIFilename` `\SetTEIFilename` $\langle basename \rangle$  is a preamble-only command. It is used to set the base name of the TEI `xml` output file, to which the suffix `.xml` is appended. By default, the base name is `\jobname-tei`:—

```

250 \NewDocumentCommand{\SetTEIFileName}{m}{
251   \luadirect{ekdosis.setteifilename(\luastringN{#1})}
252 }
253 \@onlypreamble\SetTEIFileName

```

`\AddxmlBibResource` This is a preamble-only command. If a base name (either suffixed with `.xml` or not) for a TEI `xml`-compliant bibliographical database file be provided with `\AddxmlBibResource` $\langle basename or name.xml \rangle$ , `ekdosis` will use it and insert formatted data in the back matter section of its own TEI `xml` output file, as `<biblStruct>` elements within a `<listBibl>` section.

```

254 \NewDocumentCommand{\AddxmlBibResource}{m}{
255   \luadirect{ekdosis.addxmlbibresource(\luastringN{#1})}
256 }
257 \@onlypreamble\AddxmlBibResource

```

`\ekd@test@lang` `\ekd@test@lang` is used internally by `ekdosis`. This command returns `\ekd@lang@pkgtrue` if either `babel` or `polyglossia` be used so that `\languagenam` can be inserted when and where needed in the apparatus criticus.

```

258 \newif\ifekd@lang@pkg
259 \NewDocumentCommand{\ekd@test@lang}{}{%
260   \ltx@ifpackageloaded{babel}{\ekd@lang@pkgtrue}{}%
261   \ltx@ifpackageloaded{polyglossia}{\ekd@lang@pkgtrue}{}%
262 }

```

**Multiple-layer apparatuses** ekdosis must know if an entry is to be processed in a single- or multiple-layer context:—

```
263 \newif\ifekd@mapps
```

Now the key-value options can be defined:—

```
264 \ekvdefinekeys{ekd@newapp}{
265   choice direction = {LR = \def\direction@val{LR},
266                      RL = \def\direction@val{RL}},
267   unknown-choice direction = \PackageError{ekdosis}{unknown
268     direction=#1}{`direction' must be either `LR' or `RL'.},
269   store rule = \rule@val,
270   nmeta norule = {rule=none},
271   code delim = \def\delim@val{\unexpanded{#1}},
272   store sep = \sep@val,
273   store subsep = \subsep@val,
274   store bhook = \bhook@val,
275   store ehook = \ehook@val,
276   store maxentries = \limit@val,
277   store lang = \lang@val,
278   store notelang = \notelang@val,
279   initial direction = LR,
280   initial delim = {},
281   initial ehook = {\csname ekd@end@apparatus\endcsname}
282 }
```

`\DeclareApparatus` `\DeclareApparatus{<apparatus name>}[<options>]` is a preamble-only command. As a mandatory argument, it takes the name of the new layer of notes to be inserted in the apparatus block. Then, the following seven key-value options can be used to lay out the layer: `direction=LR|RL`, `rule`, `delim` (the delimiter between entries), `sep` (the separator between lemma part and readings or notes), `bhook` ( $\LaTeX$  code inserted as the layer begins), `ehook` ( $\LaTeX$  code inserted as the layer ends), `maxentries` (if set and `maxentries >= 10`, the number of entries at which a `\pagebreak` is issued):—

```
283 \NewDocumentCommand{\DeclareApparatus}{m O{}}{
284   \newbool{subsq@unit@#1}
285   \booltrue{subsq@unit@#1}
286   \unless\ifekd@mapps\global\ekd@mappstrue\fi
287   \bgroup
288   \ekvset{ekd@newapp}{#2}
289   \luadirect{ekdosis.newapparatus(
290     \luastringN{#1},
291     \luastring{\direction@val},
292     \luastringO{\rule@val},
293     \luastringO{\delim@val},
294     \luastringO{\sep@val},
295     \luastringO{\subsep@val},
296     \luastringO{\bhook@val},
297     \luastringO{\ehook@val},
298     \luastringO{\limit@val},
299     \luastringO{\lang@val},
300     \luastringO{\notelang@val}
301   )}
302   \egroup
303 }
304 \@onlypreamble\DeclareApparatus
```

`\addentries` If `maxentries` be set for a given layer of critical notes, `\addentries[⟨layer⟩]{⟨n⟩}`, where `⟨n⟩` is an integer, can be used to add `⟨n⟩` to—or remove it from if `⟨n⟩` be negative—the number of accepted entries on the current page. `\addentries` operates on the default layer of notes, but any other declared layer can be specified in the optional argument of the command.

```

305 \NewDocumentCommand{\addentries}{0{\ekdan@type} m}{%
306   \luadirect{ekdosis.addto_bagunits(\luastringO{#1}, \luastringN{#2})}%
307   \ignorespaces
308 }

```

`\ekdpb` `\ekdpk[⟨page no⟩]{⟨line no⟩}` is used to insert conditional page breaks by specifying that the page break should occur only on a given line and optionally a given page. If the specified conditions be met then this command triggers `\pagebreak`.

```

309 \newcounter{ekd@pb}
310 \globalcounter{ekd@pb}
311 \NewDocumentCommand{\ekdpb}{s o m}{%
312   \IfBooleanTF{#1}
313   {\ifekd@showpb\marginpar{\ekd@hpbmk}\fi
314     \pagebreak
315     \@ifnextchar\bgroup}{-}{#3}%
316   }
317   {%
318     \def\@tmpoarg{#2}%
319     \def\@tmpmarg{#3}%
320     \stepcounter{ekd@pb}%
321     \linelabel{ekdpb:\theekd@pb}%
322     \def\tmp@ln{%
323       \getrefnumber{ekdpb:\theekd@pb}}%
324     \def\tmp@pg{%
325       \getpagerefnumber{ekdpb:\theekd@pb}}%
326     \IfNoValueTF{#2}
327     {\ifnum
328       \pdf@strcmp{\@tmpmarg}{\tmp@ln} = 0
329       \ifekd@showpb\marginpar{\ekd@spbmk}\fi
330       \pagebreak
331       \else
332       \ifekd@showpb\marginpar{[\ekd@spbmk]}\fi
333       \fi}
334     {\ifnum
335       \pdf@strcmp{\@tmpoarg}{\tmp@pg} = 0
336       \ifnum
337       \pdf@strcmp{\@tmpmarg}{\tmp@ln} = 0
338       \ifekd@showpb\marginpar{\ekd@spbmk}\fi
339       \pagebreak
340       \else
341       \ifekd@showpb\marginpar{[\ekd@spbmk]}\fi
342       \fi
343       \fi
344     }%
345   }\ignorespaces
346 }

```

Apparatus-related settings and functions. Some booleans to check if an apparatus should be inserted and what is the current environment.

```

347 \newbool{do@app}

```

```

348 \newif\ifekd@state
349 \newif\ifekd@isinapp
350 \newif\ifekd@isinlem
351 \newif\ifekd@appinapp

```

The next boolean is shared with arabluatex. \LRnum is used internally to ensure that numerals referring to line spans are displayed in the right order.

```

352 \providebool{al@rlmode}
353 \ifpackageloaded{arabluatex}{-}{%
354   \def\setRL{\booltrue{al@rlmode}\pardir TRT\textdir TRT}
355   \def\setLR{\boolfalse{al@rlmode}\pardir TLT\textdir TLT}
356 }
357 \protected\def\LRnum#1{\bgroup\textdir TLT#1\egroup}

```

Set a counter referring to line numbers and make it global.

```

358 \newcounter{ekd@lab}
359 \globalcounter{ekd@lab}

```

This command inserts words in the apparatus criticus without checking if both ekd@isinapp and ekd@state are set to true.

```

360 \NewDocumentCommand{\unconditional@appin}{o m}{%
361   \IfNoValueTF{#1}
362   {\luadirect{ekdosis.appin(\luastring0{#2})}}
363   {\luadirect{ekdosis.appin(\luastring0{#2}, \luastring0{#1})}}}%
364 }

```

`\blfootnote` `\blfootnote{footnote}` is used internally to insert the apparatus in the footnote block should the global optional argument layout be set to `footins`. Therefore, it is not documented.

```

365 \def\blfootnote{\gdef\@thefnmark{\relax}\@footnotetext}
366 % \def\blfootnote{\gdef\@thefnmark{}\@blfootnotetext}
367 \long\def\@blfootnotetext#1{\insert\footins{%
368   \reset@font\footnotesize
369   \interlinepenalty\interfootnotelinepenalty
370   \splittopskip\footnotesep
371   \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
372   \hsize\columnwidth \@parboxrestore
373   \protected@edef\@currentlabel{%
374     \csname p@footnote\endcsname\@thefnmark
375   }}%
376   \color@begingroup
377     \makeblfntext{%
378       \rule{z@footnotesep\ignorespaces#1\@finalstrut\strutbox}%
379     \color@endgroup}}%
380 \newcommand\@makeblfntext[1]{%
381   \parindent 1em%
382   \noindent
383   \hb@xt@0em{\hss\@makefnmark}#1}

```

**Single-layer apparatus** The following commands are for general settings. All of them can be used in the preamble or at any point of the document. The keys to be used follow:—

```

384 \newif\ifrtl@app
385 \def\ekdsep[] }
386 \def\ekdsubsep{}
387 \ekvdefinekeys{default@app}{

```

```

388 choice direction = {LR = \rtl@appfalse,
389   RL = \rtl@apptrue},
390 unknown-choice direction = \PackageError{ekdosis}{unknown
391   direction=#1}{\direction' must be either `LR' or `RL'.},
392 code sep = \def\ekdsep{#1},
393 code subsep = \def\ekdsubsep{#1},
394 store bhook = \ekd@begin@apparatus,
395 initial bhook = {},
396 store ehook = \ekd@end@apparatus,
397 initial ehook = {},
398 store delim = \ekd@unit@delim,
399 initial delim = {},
400 store rule = \ekd@default@rule,
401 initial rule = \rule{0.4\columnwidth}{0.4pt},
402 noval norule = \def\ekd@default@rule{\mbox{}},
403 store lang = \ekd@singleapp@lang,
404 initial lang = \ltx@ifpackageloaded{babel}{\languagename}{%
405   \ltx@ifpackageloaded{polyglossia}{\languagename}{}},
406 store notelang = \ekd@singleapp@note@lang,
407 initial notelang = \ltx@ifpackageloaded{babel}{\languagename}{%
408   \ltx@ifpackageloaded{polyglossia}{\languagename}{}}
409 }

```

`\SetApparatus` All settings can also be defined as key–value options within the argument of `\SetApparatus`:—

```

410 \NewDocumentCommand{\SetApparatus}{m}{
411   \ekvset{default@app}{#1}
412 }

```

`\SetLTRapp` `\SetLTRapp` and `\SetRTLapp` are two argument–less commands to set the direction of `\SetRTLapp` single–layer apparatus criticus, either LTR or RTL:—

```

413 \NewDocumentCommand{\SetRTLapp}{}{\rtl@apptrue}
414 \NewDocumentCommand{\SetLTRapp}{}{\rtl@appfalse}

```

`\SetSeparator` `\SetSeparator{<separator>}` allows to change the separator between lemma texts and variant readings, which is by default a closing square bracket followed by a space (`]`):—

```

415 \NewDocumentCommand{\SetSeparator}{m}{\def\ekdsep{#1}}

```

`\SetSubseparator` `\SetSubseparator{<sub-separator>}` allows to change the “subseparator” between variant readings. By default, no subseparator is set:—

```

416 \NewDocumentCommand{\SetSubseparator}{m}{\def\ekdsubsep{#1}}

```

`\SetBeginApparatus` `\SetBeginApparatus{<characters>}` can be used to append characters at the beginning of the apparatus block. By default, nothing is appended:—

```

417 \NewDocumentCommand{\SetBeginApparatus}{m}{\def\ekd@begin@apparatus{#1}}

```

`\SetEndApparatus` `\SetEndApparatus{<characters>}` can be used to append characters at the end of the apparatus block—such as a period, as it is customary in some editions. By default, nothing is appended:—

```

418 \NewDocumentCommand{\SetEndApparatus}{m}{\def\ekd@end@apparatus{#1}}

```

`\SetUnitDelimiter` `\SetUnitDelimiter{<delimiter>}` can be used to set the delimiter between entries in the apparatus criticus. By default, there is no delimiter except a simple space. `\SetUnitDelimiter` can be used to insert a broad space (with `\hskip` for instance, as in the OCT series) or the divider–sign (`||`, as in the Budé series):—

```

419 \NewDocumentCommand{\SetUnitDelimiter}{m}{\def\ekd@unit@delim{#1}}

```



`\SetApparatusLanguage` `\SetApparatusLang{(language name)}` can be used when it is needed to apply in the apparatus criticus a language different from the one that is selected in the edition text.

```
420 \NewDocumentCommand{\SetApparatusLanguage}{m}{%
421   \def\ekd@singleapp@lang{#1}}
```

`\SetApparatusNoteLanguage` `\SetApparatusNoteLang{(language name)}` can be used when it is needed to apply in entries introduced by the `\note` command a language different from the one that is selected in the edition text.

```
422 \NewDocumentCommand{\SetApparatusNoteLanguage}{m}{%
423   \def\ekd@singleapp@note@lang{#1}}
```

`\footnoteruletrue` As ekdosis takes care of drawing a rule separating the main text from the apparatus block as well as layers of notes from each other inside this block, it may not be desirable to have the standard L<sup>A</sup>T<sub>E</sub>X “footnoterule” printed on every page of the edition text. `\footnoterulefalse` removes it while `\footnoteruletrue` leaves it untouched. The latter is set by default.

```
424 \newif\iffootnoterule
425 \footnoteruletrue
426 \let\dfilt@footnoterule\footnoterule
427 \let\dfilt@pcol@footnoterule\pcol@footnoterule
428 \renewcommand\footnoterule{%
429   \iffootnoterule
430     \dfilt@footnoterule%
431   \fi
432 }
433 \renewcommand\pcol@footnoterule{%
434   \iffootnoterule
435     \dfilt@pcol@footnoterule%
436   \fi
437 }
```

`\SetDefaultRule` By default, ekdosis draws separating rules the definition of which is `\rule{0.4\columnwidth}{0.4pt}`. This can be changed in the preamble or at any point of the document with `\SetDefaultRule{(rule definition)}`. Leaving this argument empty as in `\SetDefaultRule{}` removes the rule.

```
438 \NewDocumentCommand{\SetDefaultRule}{m}{%
439   \def\@tempa{#1}
440   \ifx\@tempa\empty\def\ekd@default@rule{\mbox{}}%
441   \else%
442     \def\ekd@default@rule{#1}%
443   \fi}
```

`\NLS` `\NLS` was previously adapted from a snippet written by Heiko Oberdiek. It is used by ekdosis internally to prevent page breaks between separating rules and subsequent notes. Therefore, it is not documented.

```
444 \newcommand*{\NLS}{%
445   \nobreak\@normalcr\relax
446   % \par
447   % \nobreak
448   % \vspace{-\parskip}%
449   % \leavevmode
450   % \noindent
451   % \ignorespaces
452 }
```



This boolean is used to test if a given entry is to be preceded by a numeral referring to the line of the edition text.

```
453 \newif\ifsubs@unit
454 \subs@unittrue
```

`\add@@apparatus` inserts the apparatus block on a given page either in the footnote floating block or in a float of its own, depending on the value set in the `layout` global option. As some commands need to know whether they are called from inside the apparatus criticus, two conditionals are first defined.

```
455 \newif\ifekd@inside@app
456 \newif\ifekd@keepinapp
```

Then `\ekd@fitapp` is defined for `layout=fitapp`:—

```
457 \if@pkg@fitapp
458 \newtcboxfit{\ekd@fitapp}{%
459   blankest,
460   \if@pkg@breakable breakable\fi,
461   fit basedim = \f@size pt,
462   fit fontsize macros,
463   fit height from=0pt to \ekd@app@height,
464   fit algorithm = \ekd@fit@algorithm,
465   float=!b}
466 \fi
```

Then `\ekd@breakable` for `layout=breakable`:—

```
467 % \if@pkg@breakable
468 % \newtcboxfit{\ekd@breakable}{%
469 %   blankest,
470 %   breakable,
471 %   float=!b}
472 % \fi
```

Finally two commands are used to actually insert the apparatus depending on the value set in the `layout` global option.

```
473 \long\def\ekd@insert@apparatus{%
474 \unless\ifekd@mapps
475 \ifrtl@app\pardir TRT\leavevmode\textdir TRT\else
476 \pardir TLT\leavevmode\textdir TLT\fi
477 \fi
478 \if@pkg@parnotes
479 \if@parnotesroman
480 \renewcommand*{\theparnotemark}{\roman{parnotemark}}\fi
481 \parnoteclear\fi
482 \ekd@inside@apptrue
483 \ekd@appfontsize
484 \ifekd@mapps
485 \ifdefined\ekd@initial@rule
486 \ekd@initial@rule
487 \fi
488 \fi
489 \apparatus\unless\ifekd@mapps\ekd@end@apparatus\fi
490 \ekd@inside@appfalse
491 \if@pkg@parnotes\parnotes\parnotereset\fi
492 }%
493 \def\add@@apparatus{%
494 \if@pkg@parnotes\parnotes\else\fi
```

```

495 \if@pkg@footins
496 \bgroup
497 \unless\ifekd@mapps
498 \ifrtl@app\pardir TRT\leavevmode\textdir TRT\else
499 \pardir TLT\leavevmode\textdir TLT\fi
500 \fi
501 \blfootnote{%
502 \if@pkg@parnotes
503 \if@parnotesroman
504 \renewcommand*{\theparnotemark}{\roman{parnotemark}}\else\fi
505 \parnoteclear\else\fi
506 \ekd@inside@aptrue
507 \ekd@appfontsize
508 \ifekd@mapps
509 \ifdefined\ekd@initial@rule
510 \ekd@initial@rule
511 \fi
512 \fi
513 \apparatus\unless\ifekd@mapps\ekd@end@apparatus\fi
514 \ekd@inside@appfalse
515 \if@pkg@parnotes\parnotes\parnotereset\else\fi
516 }%
517 \egroup
518 \fi
519 \if@pkg@float
520 \begin{ekdapparatus}[!b]%
521 \ekd@insert@apparatus
522 \end{ekdapparatus}%
523 \fi
524 \if@pkg@keyfloat
525 \ekd@insert@keyparapp
526 \fi
527 \if@pkg@fitapp
528 \ekd@fitapp{\ekd@insert@apparatus}%
529 \fi
530 % \if@pkg@breakable
531 % \ekd@breakable{\ekd@insert@apparatus}%
532 % \fi
533 }

```

Before inserting any new entry, `\add@apparatus` calls `\test@apparatus` to decide whether a new apparatus block must be created on a given page.

```

534 \def\add@apparatus{%
535 \test@apparatus%
536 \ifbool{do@app}{\subsqu@unitfalse\add@@apparatus}{}%
537 }

```

`\append@app` inserts a bare (sub)entry in the apparatus...

```

538 \NewDocumentCommand{\append@app}{o +m}{%
539 \ifekd@isinapp%
540 \ifekd@state%
541 \IfNoValueTF{#1}%
542 {\luadirect{ekdosis.appin(\luastring0{#2})}}%
543 {\luadirect{ekdosis.appin(\luastring0{#2}, \luastring0{#1})}}%
544 \fi%
545 \fi}

```

while `\append@ln@app` inserts a (sub)entry possibly preceded by a line number.

```

546 \NewDocumentCommand{\append@ln@app}{o o +m}{%
547   \IfNoValueTF{#2}
548   {\IfNoValueTF{#1}
549     {\luairect{tex.sprint(ekdosis.mdvappend(\luastring0{#3}))}}
550     {\luairect{tex.sprint(ekdosis.mdvappend(\luastring0{#3},
551       \luastring0{#1}))}}}%
552   }
553   {\IfNoValueTF{#1}
554     {\luairect{tex.sprint(ekdosis.mdvappend(\luastring0{#3}, nil,
555       \luastring0{#2}))}}
556     {\luairect{tex.sprint(ekdosis.mdvappend(\luastring0{#3},
557       \luastring0{#1},
558       \luastring0{#2}))}}}%
559   }%
560 }

```

### Lineation Settings

`\outerlinenumbers` ekdosis does not use the “pagewise” numbering mode that is provided by `lineno`. Therefore, `\innerlinenumbers` `\outerlinenumbers` and `\innerlinenumbers` are defined in addition to `\rightlinenumbers` and `\leftlinenumbers`.

```

561 \def\outerlinenumbers{%
562   \def\makeLineNumberRunning{%
563     \checkoddpages
564     \ifoddpages
565       \linenumbersfont\hskip\linenumberssep\hskip\textwidth
566       \hbox to\linenumberswidth{\hss\LineNumber}\hss
567     \else
568       \hss\linenumbersfont\LineNumber\hskip\linenumberssep
569     \fi
570   }%
571 }
572 \def\innerlinenumbers{%
573   \def\makeLineNumberRunning{%
574     \checkoddpages
575     \ifoddpages
576       \hss\linenumbersfont\LineNumber\hskip\linenumberssep
577     \else
578       \linenumbersfont\hskip\linenumberssep\hskip\textwidth
579       \hbox to\linenumberswidth{\hss\LineNumber}\hss
580     \fi
581   }%
582 }

```

The keys to be used for lineation settings follow. A conditional is defined beforehand so that ekdosis may know whether the numbering should start afresh at the top of each page.

```

583 \newif\ifekd@pagelineation
584 \newif\ifekd@pagevlineation
585 \NewDocumentCommand{\ekdatbegshihook}{-}{%
586   \ifekd@pagelineation\resetlinenumber\fi
587 }
588 \AddToHook{shipout/before}{\ekdatbegshihook}
589 \newif\ifekd@elidednumbers
590 \ekvdefinekeys{ekd@lineation}{

```

```

591 choice lineation = {page = \ekd@pagelineationtrue,
592   document = \ekd@pagelineationfalse,
593   none = \ekd@pagelineationtrue
594     \renewcommand\thelinenumbers{}},
595 unknown-choice lineation = \PackageError{ekdosis}{unknown
596   lineation=#1}{`lineation' must be either `page' or `document'.},
597 choice vlineation = {page = \ekd@pagevlineationtrue,
598   document = \ekd@pagevlineationfalse},
599 unknown-choice vlineation = \PackageError{ekdosis}{unknown
600   vlineation=#1}{`vlineation' must be either `page' or `document'.},
601 code modulonum = \chardef\c@linenumbermodulo#1\relax,
602 noval modulo = \modulolinenumbers,
603 code vmodulo = \ifekd@memoir@loaded\linenumberfrequency{#1}
604   \else\if@pkg@poetry@verse\poemlines{#1}\fi\fi,
605 initial vmodulo = 1,
606 default vmodulo = 5,
607 bool vnumbrokenlines = \ifnum@brokenline,
608 bool continuousvnum = \if@continuous@vnum,
609 choice numbers = {elided = \ekd@elidednumberstrue,
610   full = \ekd@elidednumbersfalse},
611 unknown-choice numbers = \PackageError{ekdosis}{unknown
612   numbers=#1}{`numbers' must be either `elided' or `full'.},
613 initial numbers = elided,
614 choice margin = {right = \rightlinenumbers,
615   left = \leftlinenumbers,
616   inner = \innerlinenumbers,
617   outer = \outerlinenumbers},
618 unknown-choice margin = \PackageError{ekdosis}{unknown
619   margin=#1}{`margin' must be either `left', `right', \MessageBreak
620   `inner' or `outer'},
621 choice vmargin = {
622   right = \if@pkg@poetry@verse\verselinenumbersright\fi,
623   left = \if@pkg@poetry@verse\verselinenumbersleft\fi},
624 unknown-choice vmargin = \PackageError{ekdosis}{unknown
625   vmargin=#1}{`margin' must be either `left' or `right'}
626 }

```

`\SetLineation` Then `\SetLineation{<options>}` can be used in the preamble or at any point of the document to set lineation preferences. Its argument processes the key-value options that are defined just above.

```

627 \NewDocumentCommand{\SetLineation}{m}{
628   \ekvset{ekd@lineation}{#1}
629 }

```

`\modulolinenumbers`

```

630 \NewDocumentCommand{\modulolinenumbers}{0{5}}{%
631   \ifekd@memoir@loaded
632     \linenumberfrequency{#1}%
633   \else
634     \if@pkg@poetry@verse
635       \poemlines{#1}%
636     \fi
637   \fi
638   \ignorespaces
639 }

```

Use `\normalfont` for line numbers:—

```
640 \renewcommand\linenumberfont{\normalfont\footnotesize}
```

`\SetDefaultApparatus` By default, `ekdosis` defines one layer of critical notes which is called `default`. This name can be changed at any point of the document with `\SetDefaultApparatus{<name>}`.

```
641 \ekvdefinekeys{appnote}{  
642   store type = \ekdan@type,  
643   initial type = default  
644 }  
645 \NewDocumentCommand{\SetDefaultApparatus}{m}{%  
646   \ekvset{appnote}{type=#1}}
```

`\app` `\app[type=<type>]{<apparatus entries>}` takes one mandatory argument and accepts one optional argument. `type=` refers to the layer the note must go into and `<apparatus entries>` contains command used to insert the entries, either `\lem`, `\rdg` or `\note<*>`:—

```
647 \NewDocumentCommand{\app}{0} > { \TrimSpaces } +m}{%  
648   \leavevmode  
649   \begingroup  
650   \ekvset{appnote}{#1}%  
651   \ifekd@isinapp\ekd@appinapptrue\fi  
652   \ekd@isinapptrue  
653   \stepcounter{ekd@lab}%  
654   \zlabel{ekd:\theekd@lab}%  
655   \luadirect{ekdosis.storeabspg(  
656     \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}%  
657   \ifekd@state\add@apparatus\fi  
658   \luadirect{tex.sprint(ekdosis.removeesp(\luastringN{#2}))}%  
659   \ekd@isinappfalse  
660   \ekd@appinappfalse  
661   \endgroup}
```

`\ekdpage` Instead of absolute page numbers, `ekdosis` now marks the entries of the apparatus with its own page numbering scheme. `\ekdpage` can be used at any point of the document to retrieve and print the current number.

```
662 \NewDocumentCommand{\ekdpage}{}{%  
663   \luadirect{tex.sprint(ekdosis.getekdabspg())}%  
664 }
```

`\current@ref@arg` is used outside `\app` by `\note`. It takes two mandatory arguments: the beginning line label and the ending line label—which are manually inserted—and returns the formatted line reference to be inserted in the apparatus criticus.

```
665 \def\current@ref@arg#1#2{%\textdir TLT%  
666   \unexpanded\expandafter{\ekd@refnumstyle}%  
667   \ifnum%  
668     \pdf@strcmp{\getpagerefnnumber{#1}}{\getpagerefnnumber{#2}}  
669     =  
670     0  
671     \ifnum%  
672       \pdf@strcmp{\getrefnumber{#1}}{\getrefnumber{#2}}  
673       =  
674       0  
675       %  
676       \ifekd@mapp%  
677       \ifbool{subsq@unit@\ekdan@type}{%
```

```

678     \ifnum%
679     \pdf@strcmp{\getrefnumber{#1}}{%
680       \getrefnumber{\luairect{tex.sprint(ekdosis.getprevnotelab())}}
681     =
682     0
683     \else
684     \LRnum{\getrefnumber{#1}}\unexpanded\expandafter{\ekd@postrefnum}% issue the no
685     \fi%
686   }%
687   {\LRnum{\getrefnumber{#1}}\unexpanded\expandafter{\ekd@postrefnum}}% issue the no
688   \else
689   \ifsubsqu@unit%
690   %
691   \ifnum%
692   \pdf@strcmp{\getrefnumber{#1}}{%
693     \getrefnumber{\luairect{tex.sprint(ekdosis.getprevnotelab())}}
694   =
695   0
696   \else
697   \LRnum{\getrefnumber{#1}}\unexpanded\expandafter{\ekd@postrefnum}% issue the no
698   \fi
699   %
700   \else
701   \LRnum{\getrefnumber{#1}}\unexpanded\expandafter{\ekd@postrefnum}% issue the no
702   \fi
703   \fi
704   %
705   \else
706   \ifekd@elidednumbers
707   \luairect{tex.sprint(ekdosis.numrange(\luastring{\getrefnumber{#1}},
708     \luastring{\getrefnumber{#2}}))}%
709     \unexpanded\expandafter{\ekd@postrefnum}% issue the nos
710   \else
711   \LRnum{\getrefnumber{#1}}--%
712   \LRnum{\getrefnumber{#2}}\unexpanded\expandafter{\ekd@postrefnum}% issue the nos
713   \fi
714   \fi%
715   \else
716   \ifboolexpr{bool {ekd@pagelineation} or bool {ekd@pagevlineation}}
717     {\LRnum{\getrefnumber{#1}}--%
718     \LRnum{\getpagerefnnumber{#2}}.%
719     \LRnum{\getrefnumber{#2}}\unexpanded\expandafter{\ekd@postrefnum}}% issue pg and ln nos
720     {\LRnum{\getrefnumber{#1}}--%
721     \LRnum{\getrefnumber{#2}}\unexpanded\expandafter{\ekd@postrefnum}}% issue the nos
722     \fi%
723     \ifekdn@forcenum
724     \LRnum{\getrefnumber{#1}}\unexpanded\expandafter{\ekd@postrefnum}% force the no
725     \fi
726   }%
727 }

```

\current@ref is pretty much the same as \current@reg@arg, but takes no argument. It is used by commands such as \lem when references to page and line numbers can be returned by Lua.

```

728 \def\current@ref{%\textdir TLT%
729   \unexpanded\expandafter{\ekd@refnumstyle}%

```

```

730 \ifnum%
731 \pdf@strcmp{%
732 \getpagerefnnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
733 {\getpagerefnnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-e}}
734 =
735 0
736 \ifnum%
737 \pdf@strcmp{%
738 \getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
739 {\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-e}}
740 =
741 0
742 %
743 \ifekd@mapps%
744 \ifbool{subsq@unit@ekdan@type}{%
745 \ifnum%
746 \pdf@strcmp{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
747 {\getrefnumber{\luairect{tex.sprint(ekdosis.getprevlnlab())}-b}}
748 =
749 0
750 %%%begin
751 \ifnum%
752 \pdf@strcmp{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-e}}%
753 {\getrefnumber{\luairect{tex.sprint(ekdosis.getprevlnlab())}-e}}
754 =
755 0
756 \ifekd@appinapp
757 \ifnum%
758 \pdf@strcmp{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
759 {\getrefnumber{\luairect{tex.sprint(ekdosis.getprevprevlnlab())}-b}}
760 =
761 0
762 \else
763 \LRnum{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
764 \unexpanded\expandafter{\ekd@postrefnum}% issue the no
765 \fi
766 \fi
767 \else
768 \LRnum{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
769 \unexpanded\expandafter{\ekd@postrefnum}% issue the no
770 \fi
771 %%% end
772 \else
773 \LRnum{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
774 \unexpanded\expandafter{\ekd@postrefnum}% issue the no
775 \fi%
776 }{\LRnum{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
777 \unexpanded\expandafter{\ekd@postrefnum}}% issue the no
778 \else
779 \ifsubsq@unit%
780 %
781 \ifnum%
782 \pdf@strcmp{\getrefnumber{\luairect{tex.sprint(ekdosis.getlnlab())}-b}}%
783 {\getrefnumber{\luairect{tex.sprint(ekdosis.getprevlnlab())}-b}}
784 =

```

```

785 0
786 %%%begin
787 \ifnum%
788 \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}%
789 {\getrefnumber{\luadirect{tex.sprint(ekdosis.getprevlnlab())}-e}}
790 =
791 0
792 \ifekd@ppinapp
793 \ifnum%
794 \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}%
795 {\getrefnumber{\luadirect{tex.sprint(ekdosis.getprevprevlnlab())}-b}}
796 =
797 0
798 \else
799 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}%
800 \unexpanded\expandafter{\ekd@postrefnum}% issue the no
801 \fi
802 \fi
803 \else
804 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}%
805 \unexpanded\expandafter{\ekd@postrefnum}% issue the no
806 \fi
807 %%% end
808 \else
809 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}%
810 \unexpanded\expandafter{\ekd@postrefnum}% issue the no
811 \fi
812 %
813 \else
814 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}%
815 \unexpanded\expandafter{\ekd@postrefnum}% issue the no
816 \fi
817 \fi
818 %
819 \else
820 \ifekd@elidednumbers
821 \luadirect{tex.sprint(ekdosis.numrange(
822 \luastring{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}},
823 \luastring{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}))}%
824 \unexpanded\expandafter{\ekd@postrefnum}% issue the nos
825 \else
826 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}--%
827 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}.%
828 \unexpanded\expandafter{\ekd@postrefnum}% issue the nos
829 \fi
830 \fi%
831 \else
832 \ifboolexpr{bool {ekd@pagelineation} or bool {ekd@pagevlineation}}
833 {\LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}--%
834 \LRnum{\getpagerefnnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}.%
835 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}%
836 \unexpanded\expandafter{\ekd@postrefnum}}% issue pg and ln nos
837 {\LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}--%
838 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}%
839 \unexpanded\expandafter{\ekd@postrefnum}}% issue the nos

```



```

840 \fi%
841 \ifekdl@forcenum
842 \LRnum{\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}}%
843 \unexpanded\expandafter{\ekd@postrefnum}% force the no
844 \fi
845 }%
846 }

```

Define keys to be used by the optional arguments of `\lem` and `\rdg`:—

```

847 \newif\ifekdl@forcenum
848 \newif\ifekdl@nonum
849 \newif\ifekdl@ilabel
850 \ekvdefinekeys{lem}{
851   code wit = \def\ekdlr@wit{#1},
852   code source = \def\ekdlr@source{#1},
853   code resp = \def\ekdlr@resp{#1},
854   code alt = \def\ekdlr@alt{#1},
855   code pre = \def\ekdlr@pre{#1},
856   code post = \def\ekdlr@post{#1},
857   code prewit = \def\ekdlr@prewit{#1},
858   code postwit = \def\ekdlr@postwit{#1},
859   code ilabel = \ekdl@ilabeltrue\def\ilabel@val{#1},
860   store type = \ekdlr@type,
861   store sep = \ekdl@sep,
862   noval nonum = \ekdl@nonumtrue,
863   noval num = \ekdl@forcenumtrue,
864   bool nodelim = \ifekdl@nodelim,
865   bool nolem = \ifekdl@nolem,
866   mmeta Nolem = {nodelim, nonum, nolem},
867   bool nosepl = \ifekdl@nosepl,
868   initial sep = \ekdsep
869 }
870 \ekvdefinekeys{rdg}{
871   code wit = \def\ekdlr@wit{#1},
872   code source = \def\ekdlr@source{#1},
873   code resp = \def\ekdlr@resp{#1},
874   code alt = \def\ekdlr@alt{#1},
875   code pre = \def\ekdlr@pre{#1},
876   code post = \def\ekdlr@post{#1},
877   code prewit = \def\ekdlr@prewit{#1},
878   code postwit = \def\ekdlr@postwit{#1},
879   store subsep = \ekdr@subsep,
880   initial subsep = \ekdsubsep,
881   bool nosubsep = \ifekdr@nosubsep,
882   store type = \ekdlr@type,
883   bool nordg = \ifekdr@nordg
884 }

```

`\rdgGrp` `\rdgGrp[(option)]{(lemma and/or readings)}` may be used to group readings so as to indicate subvariation in apparatus entries. This command is expected inside `\app{}`, and takes as argument readings to be grouped introduced by means of `\lem` and/or `\rdg` commands. It further accepts `type` as an optional key-value argument to describe the type of grouping.

```

885 \NewDocumentCommand{\rdgGrp}{0{ } > {\TrimSpaces } m}{%
886 \luadirect{tex.sprint(ekdosis.removeesp(\luastringN{#2}))}%
887 }

```

`\app@lang` `\app@lang` is used internally by `\lem` and `\rdg` to set the language for apparatus entries.  
`\app@note@lang` `\note` uses `\@note@lang`.

```

888 \def\app@lang{%
889   \ifekd@mapps
890   \luadirect{tex.sprint(ekdosis.getaplang(\luastring{\ekdan@type}))}%
891   \else
892   \ekd@singleapp@lang
893   \fi
894 }
895 \def\app@note@lang{%
896   \ifekd@mapps
897   \luadirect{tex.sprint(ekdosis.getappnotelang(\luastring{\ekdan@type}))}%
898   \else
899   \ekd@singleapp@note@lang
900   \fi
901 }

```

`\lem` `\lem[<options>]{<lemma text>}` inserts *<lemma text>* both in the edition text and in the apparatus criticus by default, preceded by the reference to the line number or a space if it is the same number as the one of the previous entry. This command accepts the optional key-value arguments just defined above.

```

902 \NewDocumentCommand{\lem}{0{} m}{%
903   \ekd@isinlemtrue
904   \bgroup
905   \ekdl@forcenumfalse
906   \ekdl@nonumfalse
907   \ekdl@ilabelfalse
908   \ekvset{lem}{#1}%
909   \ifekdl@ilabel
910   \luadirect{ekdosis.dolnlab(\luastringN{#2},
911     \luastringO{\ilabel@val})}%
912   \else
913   \luadirect{ekdosis.dolnlab(\luastringN{#2})}%
914   \fi
915   \null
916   \ekd@test@lang
917   \ifekd@mapps%
918     \ifnum%
919     \luadirect{tex.sprint(ekdosis.get_bagunits(\luastringO{\ekdan@type}))}
920     = 1
921     \boolfalse{subsq@unit@\ekdan@type}%
922     \fi%
923     \luadirect{ekdosis.increment_bagunits(\luastringO{\ekdan@type})}%
924     \def\ekd@munit@delim{%
925       \luadirect{tex.sprint(ekdosis.getappdelim(\luastringO{\ekdan@type}))}%
926       \luadirect{tex.sprint(ekdosis.limit_bagunits(\luastringO{\ekdan@type}))}%
927     \fi%
928     \ifekdl@nolem\edef\lem@app{%
929       % \hskip .75em
930       \ifekd@mapps
931       \unless\ifekdl@nodelim
932         \ifbool{subsq@unit@\ekdan@type}%
933           {\ekd@munit@delim}{-}%
934       \fi
935     \else

```

```

936 \unless\ifekdl@nodelim
937 \ifsubsqu@unit\unexpanded\expandafter{\ekd@unit@delim}\fi
938 \fi
939 \fi%
940 \unless\ifekdl@nonum\current@ref\fi%\hskip .25em}%
941 \else%
942 \ifbool{al@rlmode}{%
943 \edef\lem@app{%
944 % \hskip .75em
945 \ifekd@mapps
946 \unless\ifekdl@nodelim
947 \ifbool{subsqu@unit@ekdan@type}%
948 {\ekd@munit@delim}{}%
949 \fi
950 \else
951 \unless\ifekdl@nodelim
952 \ifsubsqu@unit\unexpanded\expandafter{\ekd@unit@delim}\fi
953 \fi
954 \fi%
955 \unless\ifekdl@nonum\current@ref\fi%\hskip .25em
956 \ifdefined\ekdlr@alt%
957 \ifdefined\ekdlr@post%
958 \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
959 {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
960 \unexpanded\expandafter{\ekdlr@alt}}%
961 \ifdefined\ekdlr@pre%
962 \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
963 \else
964 \ifdefined\ekdlr@post%
965 \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
966 {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
967 \unexpanded{#2}}%
968 \ifdefined\ekdlr@pre%
969 \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
970 \fi
971 \ifdefined\ekdlr@postwit%
972 \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
973 \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
974 \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
975 \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
976 \ifdefined\ekdlr@prewit%
977 \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
978 \ifekdl@nosep\else\unexpanded\expandafter{\ekdl@sep}\fi
979 }%
980 }%
981 {%
982 \edef\lem@app{%
983 % \hskip .75em
984 \ifekd@mapps
985 \unless\ifekdl@nodelim
986 \ifbool{subsqu@unit@ekdan@type}%
987 {\ekd@munit@delim}{}%
988 \fi
989 \else
990 \unless\ifekdl@nodelim

```

```

991     \ifsubsq@unit\unexpanded\expandafter{\ekd@unit@delim}\fi
992     \fi
993     \fi%
994     \unless\ifekdl@nonum\current@ref\fi%\hskip .25em
995     \ifdefined\ekdlr@alt%
996         \ifdefined\ekdlr@pre%
997             \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
998         \ifbool{ekd@lang@pkg}%
999             {\unexpanded\expandafter{\ekd@lemmastyle}%
1000              \noexpand\selectlanguage{\app@lang}%
1001              \unexpanded\expandafter{\ekdlr@alt}}}%
1002         {\unexpanded\expandafter{\ekd@lemmastyle}%
1003          \unexpanded\expandafter{\ekdlr@alt}}}%
1004     \ifdefined\ekdlr@post%
1005         \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
1006     \else
1007         \ifdefined\ekdlr@pre%
1008             \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
1009         \ifbool{ekd@lang@pkg}%
1010             {\unexpanded\expandafter{\ekd@lemmastyle}%
1011              \noexpand\selectlanguage{\app@lang}%
1012              \unexpanded{#2}}}%
1013             {\unexpanded\expandafter{\ekd@lemmastyle}\unexpanded{#2}}}%
1014         \ifdefined\ekdlr@post%
1015             \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
1016     \fi
1017     \ifdefined\ekdlr@prewit%
1018         \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
1019     \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
1020     \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
1021     \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
1022     \ifdefined\ekdlr@postwit%
1023         \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
1024     \ifekdl@nosep\else\unexpanded\expandafter{\ekdl@sep}\fi
1025     }%
1026 }%
1027 \fi
1028 \ifekd@mapps
1029     \ifekdl@ilabel
1030         \append@ln@app[\ekdan@type][\ilabel@val]{\lem@app}%
1031     \else
1032         \append@ln@app[\ekdan@type]{\lem@app}%
1033     \fi
1034 \else
1035     \ifekdl@ilabel
1036         \append@ln@app[][\ilabel@val]{\lem@app}
1037     \else
1038         \append@ln@app{\lem@app}%
1039     \fi
1040 \fi%
1041 \egroup%
1042 \ekd@isinlemfalse%
1043 \subsq@unittrue%
1044 }

```

`\rdg` `\rdg[⟨options⟩]{⟨variant reading⟩}` inserts *⟨variant reading⟩* in the second part of the entry, after the lemma text and the separator, in the apparatus criticus. This command accepts the optional key-value arguments defined above. This command sets `\ifekd@subsqrdrdg` to true, which instructs `ekdosis` that “subseparators” may be used for subsequent entries.

```

1045 \newif\ifekd@subsqrdrdg
1046 \NewDocumentCommand{\rdg}{0{} m}{%
1047   \bgroup%
1048   \ekvset{rdg}{#1}%
1049   \ekd@test@lang
1050   % \ifekdr@nordg\append@app{}\else% do we need \append@app{} here? If
1051   %                                     % so, keep in mind \ifekd@mapps,
1052   %                                     like so:
1053   \ifekdr@nordg
1054     \ifekd@mapps
1055       \append@app[\ekdan@type]{\ekdunspace}%
1056     \else
1057       \append@app{\ekdunspace}%
1058     \fi
1059   \else
1060   \ifbool{al@rlmode}{%
1061     \edef\rdg@app{%
1062       \ifekd@subsqrdrdg
1063         \unless\ifekdr@nosubsep\unexpanded\expandafter{\ekdr@subsep}\fi
1064       \fi
1065       \ifdefined\ekdlr@alt%
1066         \ifdefined\ekdlr@post%
1067           \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
1068         {\textdir TRT\unexpanded\expandafter{\ekd@readingstyle}%
1069           \unexpanded\expandafter{\ekdlr@alt}}%
1070       \ifdefined\ekdlr@pre%
1071         \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
1072     \else
1073       \ifdefined\ekdlr@post%
1074         \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
1075       {\textdir TRT\unexpanded\expandafter{\ekd@readingstyle}%
1076         \unexpanded{#2}}%
1077       \ifdefined\ekdlr@pre%
1078         \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
1079     \fi
1080     \ifdefined\ekdlr@postwit%
1081       \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
1082     \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
1083     \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
1084     \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
1085     \ifdefined\ekdlr@prewit%
1086       \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
1087   }%
1088 }%
1089 {%
1090   \edef\rdg@app{%
1091     \ifekd@subsqrdrdg
1092       \unless\ifekdr@nosubsep\unexpanded\expandafter{\ekdr@subsep}\fi
1093     \fi
1094     \ifdefined\ekdlr@alt%
1095     \ifdefined\ekdlr@pre%

```

```

1096     \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
1097 \ifbool{ekd@lang@pkg}%
1098 {\unexpanded\expandafter{\ekd@readingstyle}%
1099   \noexpand\selectlanguage{\app@lang}%
1100   \unexpanded\expandafter{\ekdlr@alt}}}%
1101 {\unexpanded\expandafter{\ekd@readingstyle}%
1102  \unexpanded\expandafter{\ekdlr@alt}}}%
1103 \ifdefined\ekdlr@post%
1104   \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
1105 \else
1106   \ifdefined\ekdlr@pre%
1107     \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
1108   \ifbool{ekd@lang@pkg}%
1109     {\unexpanded\expandafter{\ekd@readingstyle}%
1110      \noexpand\selectlanguage{\app@lang}\unexpanded{#2}}}{%
1111      {\unexpanded\expandafter{\ekd@readingstyle}\unexpanded{#2}}}%
1112   \ifdefined\ekdlr@post%
1113     \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
1114   \fi
1115   \ifdefined\ekdlr@prewit%
1116     \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
1117   \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
1118   \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
1119   \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
1120   \ifdefined\ekdlr@postwit%
1121     \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
1122   }%
1123 }%
1124 \ifekd@mapps
1125   \append@app[\ekdan@type]{\rdg@app}%
1126 \else
1127   \append@app{\rdg@app}%
1128 \fi
1129 \fi
1130 \egroup
1131 \ekd@subsq@rdgtrue
1132 }

```

Define keys to be used by the optional argument of `\note` when this command is found outside `\app`:—

```

1133 \newif\ifekdn@forcenum
1134 \ekvdefinekeys{note}{
1135   store type = \ekdan@type,
1136   store lem = \ekdn@lem,
1137   code labelb = \def\ekdn@labelb{#1},
1138   code labelc = \def\ekdn@labelc{#1},
1139   bool nodelim = \ifekdn@nodelim,
1140   store sep = \ekdn@sep,
1141   bool nosepl = \ifekdn@nosepl,
1142   initial type = default,
1143   initial sep = \ekdsep,
1144   bool nonum = \ifekdn@nonum,
1145   noval num = \ekdn@forcenumtrue
1146 }

```

`\note@noapp` is used internally when a `\note` command is found outside `\app`. This command is mostly used to insert short comments or references to texts quoted or cited in the edition text to go into additional layers of the apparatus criticus, e.g. the *apparatus testium*. It accepts the optional key-value arguments just defined above. It must be noted that `labelb` must be specified; otherwise `ekdosis` will issue an error message.

```

1147 \NewDocumentCommand{\note@noapp}{0{} +m}{%
1148   \leavevmode
1149   \bgroup%
1150   \ekvset{note}{#1}%
1151   \ekd@test@lang
1152   \stepcounter{ekd@lab}%
1153   \zlabel{ekd:\theekd@lab}%
1154   \luadirect{ekdosis.storeabspg(
1155     \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}%
1156   \ifekd@state\add@apparatus\fi%
1157   \ifekd@mapps%
1158     \ifnum%
1159       \luadirect{tex.sprint(ekdosis.get_bagunits(\luastring0{\ekdan@type}))}
1160       = 1
1161       \boolfalse{subsq@unit@\ekdan@type}%
1162     \fi%
1163   \luadirect{ekdosis.increment_bagunits(\luastring0{\ekdan@type})}%
1164   \def\ekd@munit@delim{%
1165     \luadirect{tex.sprint(ekdosis.getappdelim(\luastring0{\ekdan@type}))}%
1166     \luadirect{tex.sprint(ekdosis.limit_bagunits(\luastring0{\ekdan@type}))}%
1167   \fi%
1168   \ifdefined\ekdn@labelb%
1169     \luadirect{tex.sprint(ekdosis.setnotelab(\luastring0{\ekdn@labelb}))}%
1170     \ifdefined\ekdn@labela\else\def\ekdn@labela{\ekdn@labelb}\fi%
1171   \else\PackageError{ekdosis}{missing labelb}{`labelb' must be
1172   set.}\fi%
1173   \ifbool{al@rlmode}%
1174   {\edef\note@contents{%
1175     % \hskip .75em
1176     \ifekd@mapps
1177     \unless\ifekdn@nodelim
1178       \ifbool{subsq@unit@\ekdan@type}%
1179         {\ekd@munit@delim}{}%
1180     \fi
1181     \else
1182       \unless\ifekdn@nodelim
1183         \ifsubsq@unit\unexpanded\expandafter{\ekd@unit@delim}\fi
1184       \fi
1185     \fi%
1186     \unless\ifekdn@nonum\current@ref@arg{\ekdn@labelb}{\ekdn@labela}\fi\hskip .25em
1187     \ifdefined\ekdn@lem%
1188     {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
1189     \unexpanded\expandafter{\ekdn@lem}}%
1190     \unless\ifekdn@nosep
1191     \unexpanded\expandafter{\ekdn@sep}\fi
1192     \else\fi%
1193     {\textdir TRT\unexpanded{#2}}}%
1194   {\edef\note@contents{%
1195     % \hskip .75em
1196     \ifekd@mapps

```

```

1197     \unless\ifekdn@nodelim
1198       \ifbool{subsq@unit@\ekdan@type}%
1199         {\ekd@munit@delim}{}%
1200     \fi
1201 \else
1202   \unless\ifekdn@nodelim
1203     \ifsubsq@unit\unexpanded\expandafter{\ekd@unit@delim}\fi
1204   \fi
1205 \fi%
1206 \unless\ifekdn@nonum\current@ref@arg{\ekdn@labelb}{\ekdn@labelc}\fi\hskip .25em
1207 \ifdefined\ekdn@lem
1208   \ifbool{ekd@lang@pkg}%
1209   {{\unexpanded\expandafter{\ekd@lemmastyle}%
1210     \noexpand\selectlanguage{\app@lang}%
1211     \unexpanded\expandafter{\ekdn@lem}}}%
1212   {\unexpanded\expandafter{\ekd@lemmastyle}%
1213     \unexpanded\expandafter{\ekdn@lem}}}%
1214   \unless\ifekdn@nosep
1215     \unexpanded\expandafter{\ekdn@sep}\fi
1216   \else\fi%
1217   \ifbool{ekd@lang@pkg}%
1218   {{\noexpand\selectlanguage{\app@note@lang}\unexpanded{#2}}}%
1219   {\unexpanded{#2}}}%
1220 \ifekd@mapps%
1221 \unconditional@appin[\ekdan@type]{\note@contents}%
1222 \else%
1223 \unconditional@appin{\note@contents}%
1224 \fi%
1225 \luadirect{\ekdosis.setprevnotelab(\luastring0{\ekdn@labelb})}%
1226 \egroup
1227 \subsq@unittrue
1228 \ignorespaces
1229 }

```

Define keys to be used by the optional argument of `\note` when this command is found inside `\app`:—

```

1230 \ekvdefinekeys{ekd@note}{
1231   store pre = \pre@value,
1232   store post = \post@value,
1233   nmeta sep = {post=\ekdsep},
1234   nmeta subsep = {pre=\ekdsubsep}
1235 }

```

The following three commands, `\note@app`, `\ekd@note` and `\ekd@note@star` are used internally when a `\note` command is found inside `\app`. These commands are used to insert short comments after the lemma text or after any variant reading in the apparatus criticus. `\note@app` and subsequently `\ekd@note` and `\ekd@note@star` accept the optional key-value arguments just defined above.

```

1236 \NewDocumentCommand{\ekd@note}{0{} m}{%
1237   \bgroup
1238   \ekvset{ekd@note}{#1}%
1239   \edef\note@contents{%
1240     \ekvifdefinedNoVal{ekd@note}{pre}{-}{%
1241       \unexpanded\expandafter{\pre@value}}%
1242     \unexpanded{#2}%
1243     \ekvifdefinedNoVal{ekd@note}{post}{-}{%

```



```

1244     \unexpanded\expandafter{\post@value}}%
1245 }%
1246 \ifekd@mapps%
1247 \append@app[\ekdan@type]{\note@contents}%
1248 \else%
1249 \append@app{\note@contents}%
1250 \fi%
1251 \egroup
1252 }
1253 \NewDocumentCommand{\ekd@note@star}{0{} m}{%
1254 \bgroup
1255 \ekvset{ekd@note}{#1}%
1256 \edef\note@contents{%
1257   \ekvifdefinedNoVal{ekd@note}{pre}{-}{%
1258     \unexpanded\expandafter{\pre@value}}%
1259   \if@pkg@parnotes
1260     \unskip\noexpand\parnote{\unexpanded{#2}}%
1261   \else
1262     \unskip\noexpand\footnote{\unexpanded{#2}}%
1263   \fi
1264   \ekvifdefinedNoVal{ekd@note}{post}{-}{%
1265     \unexpanded\expandafter{\post@value}}%
1266 }%
1267 \ifekd@mapps
1268 \append@app[\ekdan@type]{\note@contents}%
1269 \else
1270 \append@app{\note@contents}%
1271 \fi
1272 \egroup
1273 }
1274 \NewDocumentCommand{\note@app}{s 0{} +m}{%
1275 \ifbool{al@rlmode}{%
1276   \IfBooleanTF{#1}{\ekd@note@star[#2]{%
1277     {\textdir TRT#3}}%
1278   {\ekd@note[#2]{\textdir TRT#3}}}%
1279 }{%
1280   \IfBooleanTF{#1}{\ekd@note@star[#2]{#3}%
1281     {\ekd@note[#2]{#3}}%
1282   }%
1283 }

```

`\note` Finally, `\note` is a simple command designed to check whether `\note` itself is called inside or outside `\app`. Then, unless it is found inside `\lem`, it calls `\note@app` in the former case and `\note@noapp` in the latter case:—

```

1284 \NewDocumentCommand{\note}{s 0{} +m}{%
1285   \ifekd@state%
1286     \ifekd@isinapp%
1287       \ifekd@isinlem%
1288         \note@noapp[#2]{#3}%
1289       \else%
1290         \IfBooleanTF{#1}{\note@app*[#2]{#3}}{\note@app[#2]{#3}}%
1291       \fi%
1292     \else%
1293       \note@noapp[#2]{#3}%
1294     \fi%

```

```
1295 \fi%
1296 }
```

**Emendations and Conjectures** Here follows the key-value options to be used by `\SetCritSymbols` below:—

```
1297 \ekvdefinekeys{ekd@corr}{
1298   store suppbeg = \suppb@value,
1299   store suppend = \suppe@value,
1300   store delbeg = \delb@value,
1301   store delend = \dele@value,
1302   store sicbeg = \sicb@value,
1303   store sicend = \sice@value,
1304   store gapmark = \gapm@value,
1305   initial suppbeg = \ifbool{al@rlmode}{>}{<},
1306   initial suppend = \ifbool{al@rlmode}{<}{>},
1307   initial delbeg = \ifbool{al@rlmode}{\}{\},
1308   initial delend = \ifbool{al@rlmode}{\}{\},
1309   initial sicbeg = \dag,
1310   initial sicend = \dag,
1311   initial gapmark = ***,
1312   bool keepinapp = \ifekd@keepinapp
1313 }
```

`\supplied` `\supplied{<text>}` takes as mandatory argument the text added or supplied by conjecture.

```
1314 \NewDocumentCommand{\supplied}{m}{%
1315   \ifekd@inside@app
1316     \ifekd@keepinapp
1317       \suppb@value #1\suppe@value
1318     \else
1319       #1%
1320     \fi
1321   \else
1322     \suppb@value #1\suppe@value
1323   \fi
1324 }
```

`\surplus` `\surplus{<text>}` takes as mandatory argument the text considered by the editor to be inauthentic, but nevertheless retained between braces in the edition text as it is transmitted by all witnesses.

```
1325 \NewDocumentCommand{\surplus}{m}{%
1326   \ifekd@inside@app
1327     \ifekd@keepinapp
1328       \delb@value #1\dele@value
1329     \else
1330       #1%
1331     \fi
1332   \else
1333     \delb@value #1\dele@value
1334   \fi
1335 }
```

`\sic` `\sic{<text>}` takes as mandatory argument the text deemed by the editor to be readable but not understandable. `\sic` insert `<text>` between cruces while `\sic*` prints only one crux before `<text>`.

```

1336 \NewDocumentCommand{\sic}{s m}{%
1337   \ifekd@inside@app
1338   \ifekd@keepinapp
1339   \IfBooleanTF{#1}
1340     {\sicb@value #2}
1341     {\sicb@value #2\sice@value}%
1342   \else
1343     #2%
1344   \fi
1345 \else
1346   \IfBooleanTF{#1}
1347     {\sicb@value #2}
1348     {\sicb@value #2\sice@value}%
1349   \fi
1350 }

```

`\gap` `\gap{<options>}` indicates that some amount of text has fallen away from the entire tradition. It takes as mandatory argument a comma-separated list of options that can be used to further specify the reason for omission, the unit of measurement, the quantity and extent.

```

1351 \NewDocumentCommand{\gap}{m}{%
1352   \gapm@value
1353 }

```

`\SetCritSymbols` `\SetCritSymbols{<csv list of options>}` is used to change the symbols that ekdosis uses by default for representing emendations, lacunae, omissions, gaps and editorial deletions.

```

1354 \NewDocumentCommand{\SetCritSymbols}{m}{
1355   \ekvset{ekd@corr}{#1}
1356 }

```

## Lacunae

`\ilabel` When `\lem` has been used with the optional argument `ilabel=<label>`, `\ilabel{<label>}` must be used to mark the point where the span of text corresponding to the abbreviated lemma ends. This command is used to set the ending line number of physical lacunae in the apparatus criticus.

```

1357 \NewDocumentCommand{\ilabel}{m}{%
1358   \luadirect{tex.sprint(ekdosis.getindexedlab(\luastringN{#1}))}%
1359 }

```

`\lacunaStart`

```

\lacunaEnd 1360 \NewDocumentCommand{\lacunaStart}{0}{\ignorespaces}
1361 \NewDocumentCommand{\lacunaEnd}{0}{\ignorespaces}

```

**Apparatus Criticus** `\apparatus` is used internally by ekdosis to print the apparatus at the bottom of pages. Therefore, it is not documented, but this may change in the future for it will be possible to have apparatuses printed at other places.

```

1362 \NewDocumentCommand{\apparatus}{}{%
1363   \luadirect{tex.sprint(ekdosis.appout())}

```

The following two commands call Lua functions to check whether an apparatus should be printed on a given page and to store the current column id.

```

1364 \NewDocumentCommand{\test@apparatus}{}{%
1365   \luadirect{tex.sprint(ekdosis.testapparatus())}
1366 \NewDocumentCommand{\ekd@storecol}{}{%

```

```

1367 \luadirect{ekdosis.storecurcol(\luastring{thecolumn})}%
1368 }

```

Start and stop ekdosis:

```

1369 \NewDocumentCommand{\EkdosisOn}{}{%
1370   \ekd@statetrue}
1371 \NewDocumentCommand{\EkdosisOff}{}{%
1372   \ekd@statefalse%
1373 }

```

Neutralize unwanted commands provided by lineno within the ekdosis environment:—

```

1374 \def\ekd@setlineno{%
1375   \let\setpagewiselinenumbers\relax%
1376   \let\pagewiselinenumbers\relax%
1377   \let\endpagewiselinenumbers\relax%
1378   \let\runningpagewiselinenumbers\relax%
1379   \let\realpagewiselinenumbers\relax%
1380 }

```

`ekdosis (env.)` Finally comes the `ekdosis` environment meant to receive the edition text equipped with an apparatus criticus. This environment collects its contents and delivers it to Lua functions if a TEI xml output file be desired.

```

1381 \NewDocumentEnvironment{ekdosis}{+b}{}%
1382   \ekd@setlineno%
1383   \runninglinenumbers
1384   \EkdosisOn#1}{}%
1385   \EkdosisOff
1386   \endrunninglinenumbers%
1387   \iftei@export
1388   \luadirect{ekdosis.exporttei(\luastringN{\par #1\par })}\fi}

```

**Alignment** What follows is to arrange texts in parallel columns either on single pages or on facing pages.

Define keys to be used by the `alignment` environment:—

```

1389 \ekvdefinekeys{ekd@align}{
1390   store tcols = \tcols@num,
1391   store lcols = \lcols@num,
1392   store texts = \texts@value,
1393   store apparatus = \apparatus@value,
1394   bool paired = \ifekd@paired,
1395   choice lineation = {page = \ekd@pagelineationtrue,
1396                      document = \ekd@pagelineationfalse},
1397   unknown-choice lineation = \PackageError{ekdosis}{unknown
1398     lineation=#1}{`lineation' must be either `page' or `document'.},
1399   choice segmentation = {auto = \def\segmentation@val{auto},
1400                          noauto = \def\segmentation@val{noauto}},
1401   unknown-choice segmentation = \PackageError{ekdosis}{unknown
1402     segmentation=#1}{`segmentation' must be either `auto' or
1403     `noauto'.},
1404   bool flush = \ifekd@flushapp,
1405   initial tcols = 2,
1406   initial lcols = 1,
1407   initial texts = edition;translation,
1408   initial apparatus = edition,
1409   default segmentation = auto
1410 }

```

`\SetAlignment` `\SetAlignment{<settings>}` can be used either in the preamble or at any point of the document to set or modify the keys-value settings just defined above.

```
1411 \NewDocumentCommand{\SetAlignment}{m}{
1412   \ekvset{ekd@align}{#1}
1413 }
```

Patch `paracol` to insert a hook in `\pcol@nextpage`. This hook is used to reset line numbers on new pages.

```
1414 \patchcmd{\pcol@nextpage}{%
1415   \endgroup}{%
1416   \ifekd@pagelineation\resetlinenumber\fi
1417   \endgroup}{-}{}
```

`\EkdosisColStart` and `\EkdosisColStop` initialize columns meant to receive edition texts. These commands are used internally by `ekdosis`.

```
1418 \NewDocumentCommand{\EkdosisColStart}{-}{%
1419   \ekd@setlineno%
1420   \runninglinenumbers
1421   \ekd@storecol%
1422   \stepcounter{ekd@lab}%
1423   \zlabel{ekd:\theekd@lab}%
1424   \luadirect{%
1425     ekdosis.storeabspg(\luastring{\zref@extract{ekd:\theekd@lab}{abspage}},
1426     "pg_i")}%
1427   \ifekd@pagelineation
1428     \luadirect{tex.sprint(ekdosis.checkresetlineno())}
1429   \fi
1430 }
1431 \NewDocumentCommand{\EkdosisColStop}{-}{%
1432   \stepcounter{ekd@lab}%
1433   \zlabel{ekd:\theekd@lab}%
1434   \luadirect{%
1435     ekdosis.storeabspg(\luastring{\zref@extract{ekd:\theekd@lab}{abspage}},
1436     "pg_ii")}%
1437   \endrunninglinenumbers%
1438 }
```

`alignment` (*env.*) `\begin{alignment}[<options>]... \end{alignment}` can be used as it is provided to typeset a standard critical edition text on the left-hand pages accompanied with a translation on the right-hand pages. To that effect, it provides by default two new environments, `edition` and `translation`, to be used to typeset both texts. (Either whole texts or texts entered by paragraphs alternately.) The optional argument of `alignment` accepts the exact same key-value options as `\SetAlignment` described above. One may contrast these options with those accepted by `\SetAlignment` as “local settings”.

```
1439 \NewDocumentEnvironment{alignment}{0}{
1440 {%
1441   \ekvset{ekd@align}{#1}%
1442   \luadirect{ekdosis.mkenvdata(
1443     \luastring{\texts@value},
1444     "texts"
1445   )}
1446   \ifekd@flushapp
1447     \luadirect{ekdosis.newalignment("set")}
1448   \fi
1449   \luadirect{ekdosis.mkenvdata(
```

```

1450   \luastring{\apparatus@value}, "apparatus"
1451   )}
1452   \setrunninglinenumbers
1453   \luadirect{tex.sprint(ekdosis.mkenv())}
1454   \ifekd@paired
1455   \begin{paracol}[\lcols@num]{\tcols@num}
1456   \else
1457   \begin{paracol}[\lcols@num]*{\tcols@num}
1458   \fi
1459   }
1460   {\if@pkg@breakable\flushpage\fi
1461   \end{paracol}
1462   \iftei@export\luadirect{ekdosis.export_coldata_totei()}\fi
1463   \ifekd@flushapp
1464   \luadirect{ekdosis.newalignment("reset")}
1465   \fi
1466   \luadirect{ekdosis.flushenvdata()}
1467   \luadirect{ekdosis.flushcolnums()}
1468   }

```

**Headers and Footers** ekdosis provides a mechanism of its own for headers and footers as follows. Most of it is handled by Lua functions.

`\ekd@storemark` `\ekd@storemark` is used internally by the `mark` optional argument of `\ekddiv` described below to store marks to be printed at specific places in headers or footers.

```

1469 \NewDocumentCommand{\ekd@storemark}{m}{%
1470   \stepcounter{ekd@lab}%
1471   \label{ekd:\theekd@lab}%
1472   \luadirect{ekdosis.storehfmak(
1473     \luastring{\getpagerefnumber{ekd:\theekd@lab}},
1474     \luastringN{#1})}%
1475 }

```

`\endmark` By default, `\ekdmark` described below prints the first mark that is emitted on a given page and ignores the mark corresponding to any portion of text that may be printed between the top of the page and the point where the first mark is called. `\endmark` is an argument-less command that can be used just at the end of that portion of text to instruct ekdosis to print the last-emitted mark of the preceding page instead of the first-emitted mark of the current page.

```

1476 \NewDocumentCommand{\endmark}{}{%
1477   \stepcounter{ekd@lab}%
1478   \label{ekd:\theekd@lab}%
1479   \luadirect{ekdosis.storehfmak(
1480     \luastring{\getpagerefnumber{ekd:\theekd@lab}},
1481     "", "endmk")}%
1482   \ifdefined\xspace\xspace\fi
1483 }

```

`\edkmark` `\ekdmark` is an argument-less command called in commands used to make headers and footers where the marks stored by means of the `mark` optional argument of `\ekddiv` are to be printed.

```

1484 \NewDocumentCommand{\edkmark}{}{%
1485   \luadirect{tex.sprint(ekdosis.gethfmak(\luastring{\thepage}))}%
1486 }

```

`\ekdprintmark` `\ekdprintmark{<selector>}{<signpost>}` The signposts printed in headers and footers must be passed as second argument of `\ekdprintmark` so that `ekdosis` can remove them on pages where printing them is not desirable. `<selector>` refers to three possible symbolic letters where the first can be either H or F—for header or footer—the second E or O—for odd or even—and the third L, C or R—for left, center or right. When critical editions are laid out on “paired” facing pages as described above on page 31, the second letter, E or O, must obviously be omitted for headers and footers are the same on every facing page:—

```

1487 \ekvdefinekeys{ekd@marks}{
1488   choice mark = {HEL = \def\ekd@mk{HEL},
1489     HEC = \def\ekd@mk{HEC},
1490     HER = \def\ekd@mk{HER},
1491     HOL = \def\ekd@mk{HOL},
1492     HOC = \def\ekd@mk{HOC},
1493     HOR = \def\ekd@mk{HOR},
1494     FEL = \def\ekd@mk{FEL},
1495     FEC = \def\ekd@mk{FEC},
1496     FER = \def\ekd@mk{FER},
1497     FOL = \def\ekd@mk{FOL},
1498     FOC = \def\ekd@mk{FOC},
1499     FOR = \def\ekd@mk{FOR}};
1500   unknown-choice mark = \PackageError{ekdosis}{unknown mark=#1}{`mark'
1501     must be either `HEL', `HEC', `HER', `HOL', `HOC', `HOR', `FEL',
1502     \MessageBreak `FEC', `FER', `FOL', `FOC' or `FOR'.}
1503 }
1504 \NewDocumentCommand{\ekd@printmark}{m m}{%
1505   \bgroup
1506   \kvset{ekd@marks}{mark = #1}%
1507   \luadirect{tex.sprint(ekdosis.printmark(\luastringN{#2},
1508     \luastringO{\ekd@mk}))}%
1509   \egroup
1510 }
1511 \NewDocumentCommand{\ekdprintmark}{m m}{%
1512   \def\@tempa{#1}%
1513   \def\mk@HL{HL}\def\mk@HC{HC}\def\mk@HR{HR}%
1514   \def\mk@FL{FL}\def\mk@FC{FC}\def\mk@FR{FR}%
1515   \ifx\@tempa\mk@HL
1516     \csname ekd@printmark\endcsname{HEL}{\csname
1517       ekd@printmark\endcsname{HOL}{#2}}%
1518   \else
1519     \ifx\@tempa\mk@HC
1520       \csname ekd@printmark\endcsname{HEC}{\csname
1521         ekd@printmark\endcsname{HOC}{#2}}%
1522     \else
1523       \ifx\@tempa\mk@HR
1524         \csname ekd@printmark\endcsname{HER}{\csname
1525           ekd@printmark\endcsname{HOR}{#2}}%
1526       \else
1527         \ifx\@tempa\mk@FL
1528           \csname ekd@printmark\endcsname{FEL}{\csname
1529             ekd@printmark\endcsname{FOL}{#2}}%
1530         \else
1531           \ifx\@tempa\mk@FC
1532             \csname ekd@printmark\endcsname{FEC}{\csname
1533               ekd@printmark\endcsname{FOC}{#2}}%
1534           \else

```

```

1535 \ifx\@tempa\mk@FR
1536 \csname ekd@printmark\endcsname{FER}{\csname
1537   ekd@printmark\endcsname{FOR}{#2}}%
1538 \else
1539 \csname ekd@printmark\endcsname{#1}{#2}%
1540 \fi\fi\fi\fi\fi\fi
1541 }

```

`\ekdEOprint` To set headers and footers on “paired” facing pages, `\ekdEOprint` accepts two mandatory, self-evident arguments, like so: `\ekdEOprint{<left-hand mark>}{<right-hand mark>}`. This command uses the zero-based `abspage` counter provided by `zref-abspage`. So if the number returned by this counter is odd, it falls on a left-hand page:—

```

1542 \NewDocumentCommand{\ekdEOprint}{m m}{%
1543   \ifnumodd{\theabspage}{#1}{#2}%
1544 }

```

`\ekdnofmarks` Once the signposts are marked with `\ekdprintmark`, `\ekdnofmarks` has the same effect as the L<sup>A</sup>T<sub>E</sub>X standard command `\thispagestyle{empty}`.

```

1545 \NewDocumentCommand{\ekdnofmarks}{}{%
1546   \luadirect{ekdosis.nohfmark()}%
1547 }

```

`\ekdresethmarks` `\ekdresethmarks` can be used in rare cases when it is needed to reset headers and footers to their original, viz. printable state.

```

1548 \NewDocumentCommand{\ekdresethmarks}{}{%
1549   \luadirect{ekdosis.resetfmark()}%
1550 }

```

**“Mirrored” paired pages** An easy way to have mirrored paired pages is to use a dedicated counter to set the value of the page numbers. This counter should be incremented every two pages.

`pairedpage` (*cnt.*) `pairedpage` is first set as a global counter:—

```

1551 \newcounter{pairedpage}
1552 \globalcounter{pairedpage}

```

`\setpairedpagenum` `\setpairedpagenum{<number>}` is used just ahead of the alignment environment to set the number of the first left-hand paired page.

`\setpairedpage` `\setpairedpage` is an argument-less command meant to be issued in commands used to set headers or footers before `\thepage`. This command has the counter `pairedpage` incremented on right-hand pages only, and sets `page` ← `pairedpage` on every page.

`\resetpagenumber` `\resetpagenumber` must be used right out of “mirrored” paired pages alignment environments. This argument-less command corrects any numbering error on the page following the edition text and resumes normal page numbering.

```

1553 \NewDocumentCommand{\setpairedpagenum}{m}{%
1554   \setcounter{pairedpage}{\number\numexpr#1-1}%
1555 }
1556 \NewDocumentCommand{\setpairedpage}{}{%
1557   \ifnumodd{\thepage}{\setcounter{page}{\thepairedpage}}
1558   {\stepcounter{pairedpage}\setcounter{page}{\thepairedpage}}%
1559 }
1560 \def\resetpagenumber{%
1561   \ifnumodd{\thepairedpage}{\addtocounter{page}{-1}}
1562 }

```



**Divisions of the Body** ekdosis can convert `\book`, `\part`, `\chapter`, `\section`, `\subsection` and `\subsubsection` into corresponding TEI ‘numbered’ `<divn>` elements, where  $1 \leq n \leq 6$ .

`\MkBodyDivs` `\MkBodyDivs` is used to let ekdosis know which sectional commands are actually being used in an edition text. This command takes six mandatory arguments. For example, if `\section` and `\subsection` are the only sectional commands being used, `\MkBodyDivs{section}{subsection}{-}{-}{-}{-}` will have `\section` and `\subsection` converted into `<div1>` and `<div2>` respectively.

```

1563 \NewDocumentCommand{\MkBodyDivs}{mmmmmm}{
1564   \luadirect{ekdosis.mkdivdepths(
1565     \luastringN{#1},
1566     \luastringN{#2},
1567     \luastringN{#3},
1568     \luastringN{#4},
1569     \luastringN{#5},
1570     \luastringN{#6}
1571   )}
1572 }
1573 }

```

Divisions specific to ekdosis. Define keys to be used by `\ekddiv`:—

```

1574 \ekvdefinekeys{ekd@div}{
1575   code type = \def\type@value{#1},
1576   code n = \def\n@value{#1},
1577   code head = \def\head@value{#1},
1578   code barehead = \def\barehead@value{#1},
1579   store depth = \depth@value,
1580   code mark = \ekd@storemark{#1},
1581   choice toc = {book = \def\toc@value{book},
1582                 part = \def\toc@value{part},
1583                 chapter = \def\toc@value{chapter},
1584                 section = \def\toc@value{section},
1585                 subsection = \def\toc@value{subsection},
1586                 subsubsection = \def\toc@value{subsubsection},
1587                 paragraph = \def\toc@value{paragraph},
1588                 subparagraph = \def\toc@value{subparagraph}},
1589   unknown-choice toc = \PackageError{ekdosis}{unknown toc=#1}{`toc'
1590     must be either `book', `part', `chapter', `section', `subsection',
1591     \MessageBreak `subsubsection', `paragraph' or `subparagraph'.},
1592   initial depth = 1
1593 }

```

`\FormatDiv` `\FormatDiv{<n>}{<code before>}{<code after>}` is used to lay out the heading of the title. It takes three mandatory arguments: *n*, namely the number referring to the particular depth of the division, and then some L<sup>A</sup>T<sub>E</sub>X formatting commands to go before and after the heading itself:—

```

1594 \NewDocumentCommand{\FormatDiv}{m m m}{
1595   \luadirect{ekdosis.fmtdiv(\luastring{#1},
1596     \luastringN{#2},
1597     \luastringN{#3})}
1598 }

```

`\ekd@getfmtdiv` gets the formatting commands that have been stored by `\FormatDiv`.

```

1599 \NewDocumentCommand{\ekd@getfmtdiv}{m m}{%

```

```

1600 \luairect{tex.sprint(ekdosis.getfmtdiv(\luastring0{#1},
1601 \luastringN{#2}))}%
1602 }

```

`\ekddiv` `\ekddiv{<key-value arguments>}` is the standard command provided by `ekdosis` to meet the requirements of classical and literary texts the divisions of which depend on many different received traditions. It takes one mandatory argument in which the key-value arguments defined above are accepted, and converts the divisions into TEI ‘un-numbered’ `<div>` elements.

```

1603 \NewDocumentCommand{\ekddiv}{m}{
1604 \beginngroup
1605 \ekvset{ekd@div}{#1}%
1606 \ifdefined\head@value
1607 \bgroup
1608 \ekd@getfmtdiv{\depth@value}{b}%
1609 \head@value
1610 \ekd@getfmtdiv{\depth@value}{e}%
1611 \egroup
1612 \ifdefined\toc@value
1613 \ltx@ifpackageloaded{hyperref}{\phantomsection}{}%
1614 \ifdefined\barehead@value
1615 \addcontentsline{toc}{\toc@value}{\barehead@value}%
1616 \else
1617 \addcontentsline{toc}{\toc@value}{\head@value}%
1618 \fi
1619 \fi
1620 \fi
1621 \endgroup
1622 }

```

## Poetry Settings

`ekdverse` (*env.*) `ekdverse` provides an implementation of poetry lines. It is set to use either the `lineno` or the `verse` package depending on the value that is passed to the global option `poetry`.

`\test@vpnum` `\test@vpnum` is used internally when `ekdosis` needs to know whether two subsequent lines are printed on the same page or not.

```

1623 \newif\ifekd@test@vpnum
1624 \newcounter{ekd@vpnum}
1625 \globalcounter{ekd@vpnum}
1626 \NewDocumentCommand{\test@vpnum}{}{-%
1627 \ifekd@test@vpnum
1628 \edef\@tempa{\theekd@vpnum}%
1629 \stepcounter{ekd@vpnum}%
1630 \label{vpnum:\theekd@vpnum}%
1631 \ifnum
1632 \pdf@strcmp{\getpagerefnumber{vpnum:\@tempa}}%
1633 {\getpagerefnumber{vpnum:\theekd@vpnum}}
1634 = 0
1635 \else
1636 \resetvlinenumber
1637 \fi
1638 \else
1639 \label{vpnum:\theekd@vpnum}%
1640 \global\ekd@test@vpnumtrue

```

```

1641 \fi
1642 }

```

\\+ \\+ comes in addition to the verse commands that are provided by the verse package. \\+ causes a linebreak within a verse line. In contrast to \\>, the subsequent line is not indented and complies to any already defined indent pattern. \\vscentercr must be redefined accordingly.

```

1643 \ifbool{expr{bool {@pkg@poetry@verse} or bool {ekd@memoir@loaded}}}
1644 {\newcommand{@vsifplus}[1]{\@ifnextchar +{\@firstoftwo{#1}}}
1645 \renewcommand{@vscentercr}{%
1646 \ifhmode \unskip\else \@nolnerr\fi
1647 \@vsifgt{\ifnum@brokenline@vstypelinenum\fi\verselinebreak}{%
1648 \@vsifplus{\ifnum@brokenline@vstypelinenum\fi\stepcounter{vslineno}%
1649 \par\@ifstar{\nobreak\@vsxcentercr}{%
1650 \@vsifbang{\@ifnextchar[ {\@vsicentercr}{}}{\@vsxcentercr}%
1651 }%
1652 }{%
1653 \@vstypelinenum
1654 \incr@vslineno%
1655 \par\@ifstar{\nobreak\@vsxcentercr}{%
1656 \@vsifbang{\@ifnextchar[ {\@vsicentercr}{}}{\@vsxcentercr}%
1657 }%
1658 }%
1659 }%
1660 }
1661 }{}

```

A small patch is applied to the verse package, then *ekdverse* is defined:—

```

1662 \if@pkg@poetry@verse
1663 \patchcmd{\start@vslineno}{%
1664 \ifaltindent}{%
1665 \ifekd@pagevlineation\test@vppnum\fi
1666 \ifaltindent}{}{}
1667 \ekvdefinekeys{ekd@verse}{
1668 dimen width = \vwidth@val,
1669 initial width = \linewidth,
1670 code type = \def\type@value{#1},
1671 }
1672 \ifekd@memoir@loaded
1673 \def\vlvnumfont{\normalfont\footnotesize}
1674 \def\verselinenumfont#1{\def\vlvnumfont{#1}}
1675 \else
1676 \verselinenumfont{\normalfont\footnotesize}
1677 \fi
1678 \setcounter{poemline}{1}
1679 \NewDocumentEnvironment{ekdverse}{!O}{%
1680 \ekvset{ekd@verse}{#1}%
1681 \if@continuous@vnum\setverselinenums{\thelinenumber}{0}\fi
1682 \nolinenumbers
1683 \let\linelabel\label
1684 \ifekd@memoir@loaded
1685 \refstepcounter{verse}%
1686 \else
1687 \stepcounter{verse@envctr}%
1688 \fi

```

```

1689 \addtocounter{poemline}{-1}\refstepcounter{poemline}%
1690 \setcounter{vslineno}{1}%
1691 \let\=\@vscentercr
1692 \list{}{\itemsep \z@
1693         \itemindent -\vindent%
1694         \listparindent\itemindent
1695         \parsep \stanzaskip
1696         \setlength{\itemsep}{0pt}%
1697         \setlength{\topsep}{0pt}%
1698         \setlength{\partopsep}{0pt}%
1699         \ifdim\vwidth@val < \linewidth
1700             \rightmargin \z@
1701             \setlength{\leftmargin}{\linewidth}%
1702             \addtolength{\leftmargin}{-\vwidth@val}%
1703             \addtolength{\leftmargin}{-0.5\leftmargin}%
1704         \else
1705             \rightmargin \leftmargin
1706         \fi
1707         \addtolength{\leftmargin}{\vindent}}%
1708 \item[]\ifekd@pagevlineation\test@vpnum\fi%
1709 }
1710 {\endlist
1711 \if@continuous@vnum\resetlinenumber[\thepoemline]\fi}

```

Finally, this is the standard `verse` environment:—

```

1712 \else
1713 \newlength{\ekdverseindentlength}
1714 \setlength{\ekdverseindentlength}{\parindent}
1715 \NewDocumentEnvironment{ekdverse}{!0{\ekdverseindentlength}}{
1716 \begin{list}{}{%
1717     \setlength{\leftmargin}{#1}
1718     \setlength{\itemsep}{0pt}
1719     \setlength{\topsep}{0pt}
1720     \setlength{\partopsep}{0pt}
1721 }
1722 \item[]
1723 }\end{list}}
1724 \fi

```

`\resetvlinenumber` This command is the equivalent of `\resetlinenumber` for lines of poetry. It takes an integer as optional argument, which is 1 by default.

```

1725 \NewDocumentCommand{\resetvlinenumber}{0{1}}{%
1726 \if@pkg@poetry@verse
1727 \setverselinenums{#1}{0}%
1728 \fi
1729 }

```

`ekdstanza` (*env.*) `ekdstanza` is needed when lines are grouped into stanzas, which can be further named by means of the `type` optional argument:—

```

1730 \ekvdefinekeys{ekd@stanza}{
1731 code type = \def\type@value{#1}
1732 }
1733 \NewDocumentEnvironment{ekdstanza}{!0{}}{%
1734 \leavevmode\unskip
1735 \ekvset{ekd@stanza}{#1}%
1736 \ignorespaces

```

```
1737 }{}
```

`ekdpar` (*env.*) When `autopar` is set to `false` by means of `\SetTEIxmlExport`, `ekdpar`—or any other environment set to be inserted within `<p>` elements—must be used so that `ekdosis` can be informed of paragraph boundaries.

```
1738 \NewDocumentEnvironment{ekdpar}{}{\par}{\par}
```

`\ekdunspace` Provisionally, this very simple command is used by `ekdosis` to remove undesirable spaces, notably around empty lemmas in the apparatus. As it is used internally, `\ekdunspace` is not documented.

```
1739 \newlength{\ekdspace}
1740 \settowidth{\ekdspace}{ }
1741 \def\ekdunspace{\hskip-\ekdspace}
```

## Errors and Warnings

```
1742 \def\ekd@wrong@ilabel{
1743   \PackageError{ekdosis}%
1744   {Unknown ``ilabel'' in \string\ilabel{}}%
1745   {Please check for an ``ilabel'' that exists.}
1746 }
```

**Configuration File** Finally, if a configuration file named `\jobname-ekd.cfg` can be found, this file is read and its contents loaded into the document preamble. This provides a convenient way to gather all the settings related to the critical edition in a separate file.

```
1747 \IfFileExists{\jobname-ekd.cfg}{\input{\jobname-ekd.cfg}}{}
```

## 20 Change History

v0.99a		<code>\surplus</code> : new macro for editorial deletions . . . . .	127
General: First public release (documentation in progress) . . . . .	2	<code>\TeXtoTEIPat</code> : renamed, formerly <code>\TeXtoTEIPatt</code> . . . . .	104
v1.0			
General: Documentation complete . . . . .	2	v1.2	
v1.1		<code>\ekdpb</code> : conditional page breaks . . . . .	106
<code>\addentries</code> : modifies the number of accepted entries on the current page. . . . .	105	<code>ekdstanza</code> : new environment for stanzas . . . . .	137
<code>\DeclareScholar</code> : nows builds a list of persons . . . . .	102	<code>\resetvlinenumber</code> : resets line numbers in poetry environments . . . . .	137
<code>\DeclareSource</code> : new command added . . . . .	102	<code>\SetApparatusLanguage</code> : defines an alternate language to be applied in apparatus entries . . . . .	109
<code>\gap</code> : new macro for lacunae . . . . .	128	<code>\SetHooks</code> : general hooks: new option <code>lemmastyle</code> . . . . .	100
<code>\rdgGrp</code> : new macro for grouping readings . . . . .	118	general hooks: new option <code>readingstyle</code> . . . . .	100
<code>\SetCritSymbols</code> : new macro to set the symbols to be used to mark the corrections . . . . .	128	v1.3	
<code>\sic</code> : new macro for text deemed to be not understandable . . . . .	127	<code>\edkmark</code> : prints marks in headers or footers . . . . .	131
<code>\supplied</code> : new macro for editorial additions . . . . .	127	<code>\ekdnofmarks</code> : removes headers and footers on specific pages . . . . .	133

<code>\ekdprintmark</code> : prints or removes the marks in headers and footers ...	132	<code>\vmodulolinenumbers</code> : New command to adjust modulo verse line numbering .....	113
<code>\ekdresethfmarks</code> : marks headers and footers as printable .....	133	v1.5	
<code>\ekdsetup</code> : new command for global options .....	100	<code>\eKd</code> : Prints ekdosis indentifying symbol .....	100
<code>\endmark</code> : prints the last-emitted mark of the preceding page .....	131	<code>\ekdEOprint</code> : Sets headers and footers on “paired” facing pages .....	133
<code>\SetApparatusNoteLanguage</code> : defines an alternate language to be applied in note apparatus entries .....	109	<code>\getTEIxmlid</code> : returns TEI <code>xml:ids</code> from a csv-list of ids .....	103
<code>\SetHooks</code> : general hooks: new options <code>keyparopts</code> and <code>appheight</code> .....	100	<code>\ilabel</code> : recalls and sets the ending label of lemmas used to mark lacunae in witnesses .....	128
<code>\teidirectE</code> : direct insertion of elements in the TEI <code>xml</code> file .....	103	<code>pairedpage</code> : A counter incremented every two pages .....	133
v1.4		<code>\resetpagenumber</code> : resets normal running page numbers .....	133
<code>\\+</code> : New macro for poetry .....	136	<code>\setpairedpage</code> : sets the page number of the first paired page .....	133
<code>\ekdpage</code> : prints page marks used internally by ekdosis to generate the apparatus blocks on pages .....	114	<code>\setpairedpagenum</code> : sets the same page number on paired pages ...	133
<code>\SetSubseparator</code> : New command for subseparators .....	108	<code>\teidirectE</code> : direct insertion of empty elements in the TEI <code>xml</code> file .....	103

## 21 Index

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols			
<code>\@MM</code> .....	371	<code>\@pkg@fitappttrue</code> ..	19, 20
<code>\@blfootnotetext</code> <i>366, 367</i>		<code>\@pkg@floatfalse</code> .....	17, 18, 19, 20
<code>\@currentlabel</code> .....	373	<code>\@pkg@floattrue</code> .....	16
<code>\@finalstrut</code> .....	378	<code>\@pkg@footinstrue</code> ...	17
<code>\@firstoftwo</code> .....	1644	<code>\@pkg@keyfloattrue</code> ..	18
<code>\@footnotetext</code> .....	365	<code>\@pkg@parnotesttrue</code>	35, 36
<code>\@ifclassloaded</code> .....	55	<code>\@pkg@poetry@versettrue</code>	31, 56
<code>\@ifnextchar</code> ....	315, 1644, 1650, 1656	<code>\@tempa</code> .....	439, 440, 1512, 1515, 1519, 1523, 1527, 1531, 1535, 1628, 1632
<code>\@ifpackageloaded</code> ..	353	<code>\@thefnmark</code>	365, 366, 374
<code>\@ifstar</code> .....	1649, 1655	<code>\@tmpmarg</code> ..	319, 328, 337
<code>\@makeblfntext</code> ..	377, 380	<code>\@tmpoarg</code> .....	318, 335
<code>\@makefnmark</code> .....	383	<code>\@vscentercr</code> ..	1645, 1691
<code>\@nolnerr</code> .....	1646	<code>\@vsicentercr</code> .	1650, 1656
<code>\@normalcr</code> .....	445	<code>\@vsifbang</code> ...	1650, 1656
<code>\@onlypreamble</code> ..	111, 169, 176, 197, 202, 208, 253, 257, 304	<code>\@vsifgt</code> .....	1647
<code>\@parboxrestore</code> ....	372	<code>\@vsifplus</code> ...	1644, 1648
<code>\@parnotesromantrue</code> .	36	<code>\@vstypelinenum</code> .....	1647, 1648, 1653
<code>\@pkg@breakabletrue</code> .	20		
<code>\@pkg@ekddivfalse</code> ..	26		
<code>\@pkg@ekddivstrue</code> ...	25		
		<code>\@vsxcentercr</code> ..	1649, 1650, 1655, 1656
		<code>\\</code> .....	1691
		<code>\\+</code> .....	1643
		<code>\{</code> .....	1307, 1308
		<code>\}</code> .....	1307, 1308
		<b>A</b>	
		<code>\add@apparatus</code> .	493, 536
		<code>\add@apparatus</code> .....	534, 657, 1156
		<code>\addcontentsline</code> ....	1615, 1617
		<code>\addentries</code> .....	66, 305
		<code>addname</code> (option) .....	11
		<code>\addname@value</code> ..	181, 193
		<code>\addtocounter</code> .	1561, 1689
		<code>\AddToHook</code> .....	588
		<code>\addtolength</code> .....	1702, 1703, 1707
		<code>\AddxmlBibResource</code>	76, 254
		<code>alignment</code> (env.) .	27, 1439





<code>\ekd@postrefnum</code> .....	1003, 1065, 1069, 1094, 1100, 1102	<code>\EkdosisColStop</code> ....	1431
.. 115, 684, 687, 697, 701, 709, 712, 719, 721, 724, 764, 769, 774, 777, 800, 805, 810, 815, 824, 828, 836, 839, 843	<code>\ekdlr@post</code> . 856, 876, 957, 958, 964, 965, 1004, 1005, 1014, 1015, 1066, 1067, 1073, 1074, 1103, 1104, 1112, 1113	<code>\EkdosisOff</code> ..	1371, 1385
<code>\ekd@printmark</code> .....	1504	<code>\EkdosisOn</code> ...	1369, 1384
<code>\ekd@readingstyle</code> 118, 1068, 1075, 1098, 1101, 1109, 1111	<code>\ekdlr@postwit</code> .. 858, 878, 971, 972, 1022, 1023, 1080, 1081, 1120, 1121	<code>\ekdpage</code> .....	662
<code>\ekd@refnumstyle</code> ....	114, 666, 729	<code>ekdpar (env.)</code> ....	70, 1738
<code>\ekd@setlineno</code> .....	1374, 1382, 1419	<code>ekdpar (environment)</code> ..	69, 76, 138
<code>\ekd@singleapp@lang</code> .	403, 421, 892	<code>\ekdpb</code> .....	65, 309
<code>\ekd@singleapp@note@lang</code>	406, 423, 899	<code>\ekdpb*</code> .....	65
<code>\ekd@spbm</code> .....	105, 329, 332, 338, 341	<code>\ekdprintmark</code> ...	62, 1487
<code>\ekd@statefalse</code> ...	1372	<code>\ekdr@subsep</code> .....	879, 1063, 1092
<code>\ekd@statetrue</code> ....	1370	<code>\ekdresethmarks</code> 64, 1548	
<code>\ekd@storecol</code> .	1366, 1421	<code>\ekdsep</code> ..	36, 385, 392, 415, 868, 1143, 1233
<code>\ekd@storemark</code> 1469, 1580	<code>\ekdlr@pre</code> .. 855, 875, 961, 962, 968, 969, 996, 997, 1007, 1008, 1070, 1071, 1077, 1078, 1095, 1096, 1106, 1107	<code>\ekdsetup</code> .....	7, 103
<code>\ekd@subsqrordtrue</code> .	1131	<code>\ekdspac</code> 1739, 1740, 1741	
<code>\ekd@test@lang</code> .....	258, 916, 1049, 1151	<code>ekdstanza (env.)</code> .	47, 1730
<code>\ekd@test@vpnumtrue</code>	1640	<code>ekdstanza (environment)</code>	45, 47, 137
<code>\ekd@unit@delim</code> .....	398, 419, 937, 952, 991, 1183, 1203	<code>\ekdsbsep</code> ...	36, 386, 393, 416, 880, 1234
<code>\ekd@wrong@ilabel</code> .	1742	<code>\ekdunspace</code> .....	1055, 1057, 1739
<code>\ekdan@type</code> . 305, 642, 677, 744, 890, 897, 919, 921, 923, 925, 926, 932, 947, 986, 1030, 1032, 1055, 1125, 1135, 1159, 1161, 1163, 1165, 1166, 1178, 1198, 1221, 1247, 1268	<code>\ekdlr@resp</code> . 853, 873, 973, 1021, 1082, 1119	<code>ekdverse (env.)</code> ..	45, 1623
<code>\ekdatbegshihook</code> 585, 588	<code>\ekdlr@source</code> 852, 872, 974, 1020, 1083, 1118	<code>ekdverse (environment)</code> 7,	43, 45, 47, 135, 136
<code>\ekddiv</code> .....	59, 1603	<code>\ekdverseindentlength</code>	1713, 1714, 1715
<code>\ekdEOprint</code> ....	63, 1542	<code>\ekvdefinekeys</code> ...	15, 103, 112, 142, 177, 215, 264, 387, 590, 641, 850, 870, 1134, 1230, 1297, 1389, 1487, 1574, 1667, 1730
<code>\ekdl@forcenumfalse</code>	905	<code>\ekvifdefinedNoVal</code> ..	1240, 1243, 1257, 1264
<code>\ekdl@forcenumtrue</code> .	863	<code>\ekvoProcessLocalOptions</code>	53
<code>\ekdl@ilabelfalse</code> ...	907	<code>\ekvset</code> .....	110, 141, 154, 186, 223, 288, 411, 628, 646, 650, 908, 1048, 1150, 1238, 1255, 1355, 1412, 1441, 1506, 1605, 1680, 1735
<code>\ekdl@ilabeltrue</code> ...	859	<code>\empty</code> .....	440
<code>\ekdl@nonumfalse</code> ...	906	<code>\end</code> .....	522, 1461, 1723
<code>\ekdl@nonumtrue</code> ....	862	<code>\endcsname</code> .....	281, 374, 1516, 1517, 1520, 1521, 1524, 1525, 1528,
<code>\ekdl@sep</code> .	861, 978, 1024	<code>\EkdosisColStart</code> ..	1418
<code>\ekdlr@alt</code> .....	854, 874, 956, 960, 995, 1001,		





<code>\ifekd@state</code> . 223, 348, 540, 657, 1156, 1285	<code>\ignorespaces</code> ... 230, 231, 307, 345, 378, 451, 638, 1228, 1360, 1361, 1736	<code>lem</code> (option) ..... 41
<code>\ifekd@subsq@rdg</code> .... .. 1045, 1062, 1091	<code>\ilabel</code> ... 26, 1357, 1744	<code>\lem@app</code> ..... 928, 943, 982, 1030, 1032, 1036, 1038
<code>\ifekd@test@vpnum</code> ... ..... 1623, 1627	<code>ilabel</code> (option) ..... 14	<code>lemmastyle</code> (option) ... 34
<code>\ifekdl@forcenum</code> 841, 847	<code>\ilabel@val</code> ..... 859, 911, 1030, 1036	<code>\let</code> ... 426, 427, 1375, 1376, 1377, 1378, 1379, 1683, 1691
<code>\ifekdl@ilabel</code> ..... 849, 909, 1029, 1035	<code>\incr@vsline</code> ..... 1654	<code>lettrine</code> (package) ..... 49
<code>\ifekdl@nodelim</code> ..... .. 864, 931, 936, 946, 951, 985, 990	<code>initialrule</code> (option) .. 35	<code>\limit@val</code> ..... 276, 298
<code>\ifekdl@nolem</code> ... 865, 928	<code>\innerlinenumbers</code> ... ..... 52, 561, 616	<code>lineation</code> (option) .. 31, 52
<code>\ifekdl@nonum</code> ..... .. 848, 940, 955, 994	<code>\input</code> ..... 1747	<code>\linelabel</code> 41, 53, 321, 1683
<code>\ifekdl@nosep</code> ..... .... 867, 978, 1024	<code>\insert</code> ..... 367	<code>lineno</code> (package) ..... 51–53, 112, 129, 135
<code>\ifekdn@forcenum</code> 723, 1133	<code>institution</code> (option) ... 9	<code>\LineNumber</code> ..... .. 566, 568, 576, 579
<code>\ifekdn@nodelim</code> ..... .... 1139, 1177, 1182, 1197, 1202	<code>\institution@value</code> .. ..... 144, 160	<code>\linenumberfont</code> 53, 565, 568, 576, 578, 640
<code>\ifekdn@nonum</code> ..... .. 1144, 1186, 1206	<code>\interfootnotelinepenalty</code> ..... 369	<code>\linenumberfrequency</code> .. ..... 603, 632
<code>\ifekdn@nosep</code> ..... .. 1141, 1190, 1214	<code>\interlinepenalty</code> .. 369	<code>\linenumbersep</code> ... 53, 565, 568, 576, 578
<code>\ifekdr@nordg</code> ..... ... 883, 1050, 1053	<code>\item</code> ..... 1708, 1722	<code>\linenumberwidth</code> 566, 579
<code>\ifekdr@nosubsep</code> .... ... 881, 1063, 1092	<code>\itemindent</code> .. 1693, 1694	<code>\linewidth</code> 1669, 1699, 1701
<code>\IfFileExists</code> ..... 1747	<code>\itemsep</code> 1692, 1696, 1718	<code>\list</code> ..... 1692
<code>\iffootnoterule</code> ..... ..... 424, 429, 434		<code>\listparindent</code> ..... 1694
<code>\ifhmode</code> ..... 1646		<code>locus</code> (option) ..... 9
<code>\IfNoValueTF</code> . 326, 361, 541, 547, 548, 553		<code>\locus@value</code> .... 150, 166
<code>\ifnum</code> ..... 327, 334, 336, 667, 671, 678, 691, 730, 736, 745, 751, 757, 781, 787, 793, 918, 1158, 1631		<code>\long</code> ..... 367, 473
<code>\ifnum@brokenline</code> ... ... 607, 1647, 1648		<code>\LRnum</code> ..... 357, 684, 687, 697, 701, 711, 712, 717, 718, 719, 720, 721, 724, 763, 768, 773, 776, 799, 804, 809, 814, 826, 827, 833, 834, 835, 837, 838, 842
<code>\ifnumodd</code> 1543, 1557, 1561		<code>\ltx@ifpackageloaded</code> .. .. 260, 261, 404, 405, 407, 408, 1613
<code>\ifoddpage</code> ..... 564, 575		<code>ltxcmds</code> (package) ..... 67
<code>\ifrtl@app</code> . 384, 475, 498		<code>luabidi</code> (package) .... 53, 55
<code>\ifsubsq@unit</code> ... 453, 689, 779, 937, 952, 991, 1183, 1203		
<code>\iftei@export</code> ..... .... 13, 1387, 1462		
<code>iftex</code> (package) ..... 99		
<code>\ifx</code> ..... 440, 1515, 1519, 1523, 1527, 1531, 1535		
	<b>J</b>	
	<code>\jobname</code> ..... 1747	
	<b>K</b>	
	<code>keepinapp</code> (option) .... 23	
	<code>\kern</code> ..... 100	
	<code>keyfloat</code> (package) .... 6, 35	
	<code>\keyparbox</code> ..... 72	
	<code>keyparopts</code> (option) ... 35	
	<b>L</b>	
	<code>\label</code> .... 1471, 1478, 1630, 1639, 1683	
	<code>labelb</code> (option) ..... 41	
	<code>labeled</code> (option) ..... 41	
	<code>\lacunaEnd</code> ..... 26, 1360	
	<code>\lacunaStart</code> .... 26, 1360	
	<code>lang</code> (option) ..... 37, 38	
	<code>\lang@val</code> ..... 277, 299	
	<code>\language</code> ..... .. 404, 405, 407, 408	
	<code>layout</code> (option) ..... 6	
	<code>lcols</code> (option) ..... 30	
	<code>\lcols@num</code> 1391, 1455, 1457	
	<code>\leavevmode</code> .... 449, 475, 476, 498, 499, 648, 1148, 1734	
	<code>\leftlinenumbers</code> ... 615	
	<code>\leftmargin</code> ..... 1701, 1702, 1703, 1705, 1707, 1717	
	<code>\lem</code> ..... 14, 902	
	<b>M</b>	
	<code>\makeLineNumberRunning</code> ..... 562, 573	
	<code>margin</code> (option) ..... 52	
	<code>marginnote</code> (package) .. 71	
	<code>\marginpar</code> ..... 313, 329, 332, 338, 341	
	<code>\marginparthreshold</code> . 33	
	<code>mark</code> (option) ..... 60	
	<code>mmaxentries</code> (option) . 35, 39	
	<code>\mbox</code> ..... 402, 440	
	<code>memoir</code> (class) ..... 7, 44	



type	.....	13, 15, 16, 20, 40, 45, 47, 59
unit	.....	24
vlineation	.....	52
vmargin	.....	52
vmodulo	.....	52
vnumbrokenlines	..	52
width	.....	45
wit	.....	14, 16, 26
origDate (option)	.....	9
\origDate@value	..	149, 165
\outerlinenumbers	...	..... 52, <u>561</u> , 617
<b>P</b>		
\PackageError	....	22, 29, 32, 38, 49, 132, 219, 267, 390, 595, 599, 611, 618, 624, 1171, 1397, 1401, 1500, 1589, 1743
\pagebreak	..	314, 330, 339
\pagewiselinenumbers		1376
paired (option)	.....	31
pairedpage (counter)		<u>1551</u>
\par	.....	446, 1388, 1649, 1655, 1738
paracol (package)	..	32, 130
\pardir	....	354, 355, 475, 476, 498, 499
\parindent	....	381, 1714
\parnote	.....	1260
\parnoteclear	...	481, 505
\parnotereset	...	491, 515
\parnotes	..	491, 494, 515
parnotes (option)	.....	7
parnotes (package)	...	7, 16
\parsep	.....	1695
\parskip	.....	448
\partopsep	...	1698, 1720
\patchcmd	....	1414, 1663
patverse (environment)	..	47
\pcol@footnoterule	..	..... 427, 433
\pcol@nextpage	....	1414
\pdf@strcmp	....	328, 335, 337, 668, 672, 679, 692, 731, 737, 746, 752, 758, 782, 788, 794, 1632
\phantomsection	...	1613
\poemlines	....	604, 635
poetry (option)	.....	7
poetry (package)	.....	82
polyglossia (package)	...	.... 37, 53–55, 104
post (option)	...	16, 16, 17
\post@value	.....	.. 1232, 1244, 1265
postrefnum (option)	...	34
postwit (option)	....	16, 16
pre (option)	....	16, 16, 17
\pre@value		1231, 1241, 1258
prewit (option)	....	16, 16
\protected	.....	357
\protected@edef	....	373
\providebool	.....	352
<b>Q</b>		
quantity (option)	.....	24
<b>R</b>		
\raise	.....	100
rawname (option)	.....	11
\rawname@value	..	178, 190
\rdg	.....	16, <u>1045</u>
\rdg@app	.....	1061, 1090, 1125, 1127
\rdgGrp	.....	20, <u>885</u>
readingstyle (option)	..	34
\realpagewiselinenumbers	.....	1379
reason (option)	.....	24
\ref	.....	53
refnumstyle (option)	..	34
\refstepcounter		1685, 1689
\relax	.....	365, 445, 601, 1375, 1376, 1377, 1378, 1379
\renewcommand	.....	.. 428, 433, 480, 504, 594, 640, 1645
repository (option)	....	9
\repository@value		145, 161
\RequireLuaTeX	.....	2
\RequirePackage	....	1, 3, 4, 57, 58, 59, 60, 61, 62, 63, 65, 69, 76, 80, 83, 84, 85, 86, 87, 88, 90, 93
\reset@font	.....	368
\resetlinenumber	....	53, 586, 1416, 1711
\resetpagenumber		64, <u>1553</u>
\resetvlinenumber	...	.... 53, 1636, <u>1725</u>
resp (option)	.....	14, 16
\rightlinenumbers	...	614
\rightmargin	..	1700, 1705
\roman	.....	480, 504
\rtl@appfalse	...	388, 414
\rtl@apptrue	....	389, 413
\rule	.....	120, 378, 401
rule (option)	.....	37, 38
\rule@val	.....	269, 292
\runninglinenumbers	..	..... 1383, 1420
\runningpagewiselinenumbers	.....	1378
<b>S</b>		
\segmentation@val	...	..... 1399, 1400
\selectlanguage	.....	1000, 1011, 1099, 1110, 1210, 1218
sep (option)	15, 17, 37, 38, 40	
\sep@val	.....	272, 294
\SetAlignment	...	32, <u>1411</u>
\SetApparatus	....	36, <u>410</u>
\SetApparatusLanguage	.....	36, <u>420</u>
\SetApparatusNoteLanguage	.....	36, <u>422</u>
\SetBeginApparatus		36, <u>417</u>
\setcolumnwidth	....	33
\setcounter		1554, 1557, 1558, 1678, 1690
\SetCritSymbols	..	23, <u>1354</u>
\SetDefaultApparatus	..	..... 38, <u>641</u>
\SetDefaultRule	..	36, <u>438</u>
\SetEndApparatus	..	36, <u>418</u>
\SetHooks	.....	34, <u>112</u>
\setlength	.....	1696, 1697, 1698, 1701, 1714, 1717, 1718, 1719, 1720
\SetLination	....	52, <u>627</u>
\setLR	.....	55, 355
\SetLTRapp	.....	36, <u>413</u>
\setpagewiselinenumbers	.....	1375
\setpairedpage	..	63, <u>1553</u>
\setpairedpagenum		64, <u>1553</u>
\setRL	.....	55, 354
\SetRTLapp	.....	36, <u>413</u>
\setrunninglinenumbers	.....	1452
\SetSeparator	....	36, <u>415</u>
\SetSubseparator	..	36, <u>416</u>
\SetTEIFilename	..	250, 253
\SetTEIFilename	..	68, <u>250</u>
\SetTEIxmlExport	..	69, <u>222</u>
settlement (option)	....	9
\settlement@value		143, 159
\settowidth	.....	1740
\SetUnitDelimiter		36, <u>419</u>

