ἔκδοσις

Typesetting TEI xml-Compliant Critical Editions

Robert Alessi
mailto:alessi@robertalessi.net

2025/05/12 – v1.5-dev

Contents

List of listings

Abstract

ekdosis is a LuaLaTeX package designed for multilingual critical editions. It can be used to typeset texts and different layers of critical notes in any direction accepted by LuaTeX. 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 LaTeX 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 OpenBSD license.

License and Disclaimer

1. More information about the OpenBSD policy to which ekdosis adheres: https://www.openbsd.org/policy.html.

```
ekdosis -- Typesetting TEI xml-Compliant Critical Editions
-----------------------------------------------------------

Copyright (c) 2020, 2021, 2022, 2023, 2024, 2025 Robert Alessi
<alessi@robertalessi.net>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

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 or https://sr.ht/~ralessi/ekdosis
- comments, feature requests, bug reports: http://www.ekdosis.org/issues.html

This release of ekdosis consists of the following source files:

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

`\eKd` *New feature* v1.5

The distinctive emblem of ekdosis is made of the three Greek letters ε, χ and δ, like so: ε$^{\chi}$δ. 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.[2]

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).[3]

The name of this package, ekdosis, derives from a Greek action noun—ἔκδοσις—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 handed 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).

2. 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.

3. Robert Alessi, "ekdosis: Using LuaLaTeX 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), jdmdh: 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.[4] 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",[5] 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

4. 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."

5. 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, *&c.*

(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 LaTeX 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.
```

1 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 ⟨*key*⟩=⟨*value*⟩. 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[6] to generate the floating environments to be used as containers for critical footnotes. This way, the keys and values provided by this package[7] 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.[8]

New feature v1.3 (d) `layout=fitapp` As described below in sect. 13.1 on page 70, "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.[9] When this option is set, the apparatus criticus grows normally until a predefined height is reached. This height is set to `0.5\textheight` by default.[10] 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 `maxlines` or `maxentries` as described below on page 57 (for `maxlines`), and on pages 34 and 38 (for `maxentries`) and in sect. 13.1 on page 70.


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

6. Brian Dunn, *The Keyfloat package* (version 2.06) [Provides a key/value interface for generating floats] (June 29, 2021), https://ctan.org/pkg/keyfloat.
7. See ibid., sect. 2.3, pp. 13–16.
8. See below, on page 34 for more information.
9. 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.
10. 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.[11]

divs `divs=ekdosis|latex` Default: ekdosis

In many occasions, LaTeX 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.[12] However, if `divs` be set to `latex`, LaTeX 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 facilites supplied by the verse package for the typesetting of lines of poetry.[13] The `ekdverse` environment must then be used instead of the `verse` environment that is provided by the verse package as described below in sect. 9.2 on page 47.

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.[14] 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 LaTeX can be allowed.[15]

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 *New feature* v1.3

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 71. By default, page breaks generated by `\ekdpb` are identified by the string `spb`—for "soft" page break—whereas those generated by `\ekdpb*` are identified by `hpb`—for "hard" page break.

11. See below on page 35 for more information.
12. See below, sect. 12 on page 63.
13. verse does not need to be set if the memoir class be used. See sect. 9.2 on page 47 for more detail.
14. See http://www.html-tidy.org.
15. See https://texfaq.org/FAQ-spawnprog for more information on how to do this.

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 an `xml:id` in the `TEI xml` output if any.[16]

(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.

(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.

16. See on page 77 for more information.

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 `<msIdentifer>` element in the `TEI` header:[17]—

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

additional `additional`=⟨*additional*⟩: Any additional information about the manuscript that can be provided as either one or more paragraphs.

New feature v1.5

In this latter case, the paragraphs must be inserted in `ekdpar` environments as described sect. 14.2 on page 77. It is then necessary to set `autopar=true` in the preamble before declaring the witnesses. If required, `autopar` can be set back to `false` in the document.[18]

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:[19]—

```
\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},
        additional={The accidents suffered by this manuscript
          predate the three series of signatures.}]
```

\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.[20]

(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]
```

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

18. For more information on this technique, see sect. 14.2 on page 76.

19. The locus specified refers to Hippocrates' *Epidemics*, Book 6.

20. See on page 77 for more information.

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

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 ipsius</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>
      <additional>The accidents suffered by this manuscript
      predate the three series of signatures.</additional>
    </msDesc></witness>
  </listWit>
</sourceDesc>
```

`\DeclareSource`
New feature v1.1

Sources `\DeclareSource{`⟨*unique label*⟩`}{`⟨*rendition*⟩`}`

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,[21] it is advisable to use Bib(LA)TEX labels in the first argument of `\DeclareSource`. Otherwise, the 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.[22]

(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: α, which includes mss. A, M, B, R, S, L and N, and β, which includes mss. T, f, U and l.

21. See below sect. 14.7 on page 83.
22. See on page 77 for more information.

Therefore, provided that all these witnesses have been already declared, here is how the two families α and β could be declared:[23]—

```
\DeclareShorthand{a}{α}{A,M,B,R,S,L,N}
\DeclareShorthand{b}{β}{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`[24] 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 this 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 scribae ipsius} & \\
  \SigLine{M}\\
  [...]
  \SigLine{N}\\
  \multicolumn{3}{c}{\emph{Familia} \getsiglum{b}}\\
```

23. These witnesses are used in the example provided below in listing 7 on page 28.
24. See above on page 8.

```
  \SigLine{T}\\
  [...]
  \SigLine{l}\\
\end{xltabular}
```

Conspectus siglorum

	Familia α	
A	*Bongarsianus* 81	s. IX–X
	A[1] *Emendationes scribae ipsius*	
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
	Familia β	
T	*Parisinus Lat.* 5764	s. XI
f	*Vindobonensis* 95	s. XII
U	*Vaticanus Lat.* 3324	s. XI
l	*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:[25]—

```
\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 37, 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

25. 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.[26] 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.

- ekdosis also provides a two-argument `\App` command which is strictly equivalent to `\app` but allows for much more flexible code folding in the emacs text editor. Code folding may be needed when readings and critical notes grow in number to a point where the edition text becomes illegible. This command is described below in sect. 13.2 on page 72, "Using emacs". (See on page 73.)

Base text and variants As can be seen in listing 1 on page 5 and the examples provided 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.[27]

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.

`nowit` `nowit=true|false`

New feature v1.5 This named argument does not need a value as it defaults to `true` if used. When used in a positive apparatus, `nowit` instructs ekdosis to remove the witnesses attached to the lemma text with the result of printing a negative entry. In a positive apparatus, this can be a way of suggesting an aberration. In a negative apparatus, `nowit=false` inserts a positive entry, which implies that the rejected variant is well attested or judged worthy of consideration as a serious alternative. For details on the method of encoding positive or negative apparatus criticus, the reader is invited to refer to the commands and options described on pages 36–37 and sect. 6.3.2 on page 38, "Laying Out Layers With The

26. See below, sect. 7.2 on page 40, "Other Notes for Comments, Sources or Testimonia" and sect. 8 on page 42, "Footnotes" for details.

27. See below on page 34 for details.

Optional Argument of `\DeclareApparatus`".[28]

`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.[29]

`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.[30]

`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 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*⟩`}`.[31] 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 25, "Lacunae".

`delim` (no-value argument)

New feature v1.5 `delim` takes no value. If used, this argument instructs to forcibly print the delimiter that ekdosis may have decided not to print in the apparatus criticus before the lemma text.

`nodelim` (no-value argument)

New feature v1.5 `nodelim` takes no value. If used, this argument 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 page 17, 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`

28. From version 1.5 onwards, ekdosis features the option of producing negative critical apparatus either in print, in `TEI xml` output, or in both. It is therefore possible to leave the reading of the lemma text to be understood, while having all the corresponding witnesses inserted in the `TEI xml` output file. It is up to the editor to decide on the most appropriate form. These new functions were added as a result of a discussion with Orsola Lorena Purcaro during a set of seminars given by the author while in residence at the University Federico II of Naples in April 2025. Special thanks to her.

29. For edition texts used as sources, see examples below in sect. 3 on page 20 and sect. 14.7 on page 83.

30. See detailed examples in sect. 3 on page 20.

31. See below on page 26.

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 25, "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.

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

`pre` `pre=`⟨*words*⟩

`pre` inserts ⟨*words*⟩ *before* the lemma text.

`post` `post=`⟨*words*⟩

`post` inserts ⟨*words*⟩ *after* the lemma text.

`prewit` `prewit=`⟨*words*⟩

`prewit` inserts ⟨*words*⟩ *before* the list of witnesses.

`postwit` `postwit=`⟨*words*⟩

`postwit` inserts ⟨*words*⟩ *after* the list of witnesses.

`\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 35. 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`.[32]

\note \note* **Notes** `\note[`⟨*options*⟩`]{`⟨*text*⟩`}` or `\note*[`⟨*options*⟩`]{`⟨*text*⟩`}`

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:—

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`.[33]

subsep This argument-less option is equivalent to `pre=\ekdsubsep`.[33]

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

```
\begin{ekdosis}
  I
  \app{
    \lem[wit=A]{saw}
    \rdg[wit=B]{met}}
  my friend
  \app{
    \lem{Peter}
    \rdg{John}
  }
  at the station yesterday. We were both in a
  \app{
    \lem[wit=A]{great}
    \rdg[wit=B]{good}}
  mood.
  \app{
```

32. See on pages 35–37 and sect. 6.3.1 on page 37 for details.
33. See below on page 35 for more information and listing 5 on page 24, ll. 23–5 for an illustrative example.

```
    \lem[wit=A, alt={How nice... said}]{\enquote{How nice to find
        you here!} he said.}
    \note*{There are no quotation marks in the mss.}
    \rdg[wit=B, alt=\emph{om.}]{}
  I chuckled to myself, recalling the last time we
  \app{
    \lem[wit=A,nolem]{met}
    \rdg[wit=B, alt={\emph{post} met \emph{add.} there}]{met
      there}
    \note*{Ms. \getsiglum{B} provides other additions of this kind.}}.
\end{ekdosis}
```

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

---

1 saw A] met B Peter] John great A] good B **1–2** "How nice... said A][i] *om.* B **2** *post* met *add.* there B[ii]

[i] There are no quotation marks in the mss. [ii] 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 LaTeX 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:—

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

[1] 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 appartus 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 16.

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:—

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

[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 15,[34] 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 biblatex 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;[35] 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    καὶ ἐγίνετο μᾶλλον \app{
3      \lem[wit={V, Ipc,R,H}]{νότῳ}
4      \rdg[wit=Iac]{νότου}
5      \rdg[source=Erm, type=conjecture]{ἐν νώτῳ}}· [...] % conjecture
6
7    εἰ
8    \app{
9      \lem[resp=egoscr, type=emendation]{μὲν} % emendation
10     \rdg[wit=codd, source=edd]{μὴ}
11   } εἴη διὰ ταῦτα [...]
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

34. See also on pages 10–11.

35. For detailed information on how to use `\DeclareSource` and insert references to cited works, the reader is invited to refer to sect. 14.7 on page 83.

`prewit=falso 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]{μὲν}
    \rdg[wit=codd, source=edd]{μὴ}
  } εἴη διὰ ταῦτα [...]
```

PDF output:—

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

1 νότῳ V I^{pc}RH] νότου I^{ac} ἐν νώτῳ Ermerins **2** μὲν *scripsi*] μὴ 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 νώτῳ</rdg>
</app>· [...]</p>
<p>εἰ
<app>
  <lem resp="#ego" type="emendation">μὲν</lem>
  <rdg wit="#V #I #R #H" source="#Lit #Erm #Sm">
  μὴ</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:[36]—

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

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

<*>** conjectured lacuna.

{} editorial deletion.

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

`\SetCritSymbols` *New feature* v1.1

`\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:—

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

suppbegin `suppbegin=`⟨*symbol*⟩ `Default: <`
The opening symbol used to mark the text that is supplied.

suppend `suppend=`⟨*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{}`.
New feature v1.1

`\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{}`.
New feature v1.1

`\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*⟩.
New feature v1.1

`\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:—
New feature v1.1

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:[37]—

Listing 5: Emendations, conjectures and corrections

```
% Preamble:
\DeclareShorthand{egomute}{\unskip}{ego}

% Document:
\begin{ekdosis}
  σχεδὸν \app{
    \lem[resp=egomute, nosep, post={post σχεδὸν quattuor uerba
      excidisse uid.}, type=emendation]{\supplied{\gap{reason=lost,
          unit=word, quantity=4}}}
  } οὗτοι

  subsidiis magnis \sic*{epicuri} constabilitas

  declinare quis est qui \sic{possit cernere sese}.

  \app{
    \lem[resp=egomute, type=emendation, nosep, post={ante
      ὑπογίν.}]{\surplus{καὶ}}
    \note{deleui e Gal.P}
  } ὑπογίνονται

  Πάντων δὲ \app{
    \lem[resp=egomute, type=emendation, nosep]{\supplied{τῶν πυρετῶν}}
    \note[sep]{addidi (\arb{^gamI`a 'l-.hummayAti}
      \getsiglum{Gal})}
    \rdg[nordg, source=Gal]{\arb{al-.hummayAti}}
    \rdg[wit=codd, source=edd, alt=om.]{}
  },
\end{ekdosis}
```

37. On the use of `egomute` (l. 2), see above Rem. 2 on page 22.

PDF output:—

σχεδὸν ⟨***⟩ οὗτοι
subsidiis magnis †epicuri constabilitas
declinare quis est qui †possit cernere sese†.
{καὶ} ὑπογίνονται
Πάντων δὲ ⟨τῶν πυρετῶν⟩,

1 *** post σχεδὸν quattuor uerba excidisse uid. **4** καὶ ante ὑπογίν. deleui e Gal.P **5** τῶν πυρετῶν addidi (جَمِيعَ الحُمَّيَات 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>οὗτοι</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 \lacunaEnd `\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 in 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:[38]—

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}
```

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 Ω 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 Ω 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:—

38. Cicero, *Ad Atticum epistularum libri sedecim*, recensuit H. Sjögren (Collectio scriptorum ueterum Vpsalensis; 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.

Praeteriit tempus non legis, sed libidinis tuae, fac tamen legis; ut succedatur decernitur; impedis et ais: “habe meam rationem.”

1–2 sed—legis C] *om.* Ω

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 successor. You object and cry, “Consider my candidature.”

`TEI xml` output (Latin only):—

```
<p>Praeteriit tempus non legis,
<app>
  <lem wit="#C">sed</lem>
  <rdg wit="#Ω">
    <lacunaStart />
  </rdg>
</app>libidinis tuae, fac tamen
<app>
  <lem wit="#C">legis</lem>
  <rdg wit="#Ω">
    <lacunaEnd />
  </rdg>
</app>; ut succedatur decernitur; impedis et ais:
<quote>habe meam rationem.</quote></p>
```

5 Alignment of Parallel Texts

As already said above,[39] 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.*)
`translation` (*env.*)

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

39. See point (a) on page 4.

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

`edition*` (*env.*)
`translation*` (*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.[40] See the list of sigla for manuscripts and manuscript families above on page 12. 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 91 and `TEI xml` output in sect. 17.2 on page 93. The corresponding PDF output is available in a separate file:[41]—

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

```
\begin{alignment}
  \begin{edition}
    \ekddiv{head=XIII, depth=2, n=6.13, type=section}
    In omni Gallia eorum hominum qui \app{
      \lem[wit=a]{aliquo}
      \rdg[wit=b, alt=in al-]{in aliquo}}
    sunt numero atque honore genera sunt duo. Nam plebes paene
    seruorum habetur loco, quae \app{
      \lem[wit={A,M}, alt={nihil audet (aut et \getsiglum{A1})
        per se}]{nihil audet per se}
      \rdg[wit=A1,nordg]{nihil aut et per se}
      \rdg[wit={R,S,L,N}]{nihil habet per se}
      \rdg[wit=b]{per se nihil audet}}, \app{
      \lem[wit=a]{nullo}
      \rdg[wit=b]{nulli}} adhibetur \app{
      \lem{consilio}
      \rdg[wit={T, U}, alt=conc-]{concilio}}.
  \end{edition}
  \begin{translation}
    \ekddiv{head=XIII, depth=2, n=6.13, type=section}
      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.
```

40. 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).

41. On the use of `\ekddiv` (ll. 3 and 20), see below sect. 12.2 on page 64.

```
  \end{translation}
\end{alignment}
```

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

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 undertake nothing of itself, and is admitted to no deliberation.

2 aliquo α] in al- β **4** nihil audet (aut et A¹) per se AM] nihil habet per se RSLN per se nihil audet β nullo α] nulli β **5** consilio] conc- T U

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 A¹)". 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 #l">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 #l">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 LaTeX 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 LaTeX 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.[42]

`lineation` `lineation=page|document|none` `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. `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.

`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 *Gallic 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}
    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,
```

42. For setting headers and footers in this arrangement of text, but also for the current limitations attached to this option, see on page 68.

```
  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:[43]—

```
\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.[44]

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

5.2.1 Columns and Gutters

`\columnratio` **Column Ratio on Single Pages** `\columnratio{`⟨r_1, r_2, … , r_n⟩`}`, 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{`⟨r_1, r_2, … , r_n⟩`}[`⟨r_1, r_2, … , r_n⟩`]`.

`\setcolumnwidth` **Column Width on Single Pages** `\setcolumnwidth{`⟨w_1, w_2, … , w_n⟩`}` 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.

43. The `\setmaxlines` command provides a further example of applying this technique. See below on page 57.
44. Hiroshi Nakashima, *The Paracol package* (version 1.35) [Multiple columns with texts "in parallel"] (Dec. 31, 2018), http://www.ctan.org/pkg/paracol.

Gutter Width Each value accepted by `\setcolumnwidth` can be expressed as a pair as in `\setcolumnwidth{`$\langle w_1/g_1, w_2/g_2, \ldots, w_n/g_n\rangle$`}` 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^{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/0.25 in}` 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{`$\langle w_1, w_2, \ldots, w_n\rangle$`}[`$\langle w_1, w_2, \ldots, w_n\rangle$`]` for columns only, and `\setcolumnwidth{`$\langle w_1/g_1, w_2/g_2, \ldots, w_n/g_n\rangle$`} [`$\langle w_1/g_1, w_2/g_2, \ldots, w_n/g_n\rangle$`]` for columns and gutters.

Vertical Rules Vertical rules between columns can be drawn by setting the length of the LaTeX `\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^{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.

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{`$\langle$*csv list of hooks*$\rangle$`}` 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=`$\langle$*command*$\rangle$ `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=`$\langle$*command*$\rangle$ `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, `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 50.

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 38. 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^{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. 13.1 on page 70.

nomaxentries Default: not set

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

Option Specific to the `layout=keyfloat` Global Setting[45]

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.[46] As an example, `keyparopts={ft, tr={made with ekdosis},`

45. See above (c) on page 6.
46. Dunn (cf. n. 6), sect. 2.3, pp. 13–5.

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[47]

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.

▶ To learn how this value can be adjusted on given pages, see below on page 72.

fitalgorithm `fitalgorithm=fontsize|hybrid|areasize|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.[48] While it is advisable to limit the use of fontsize to high quality typesetting for camera-ready copies, areasize 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.

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 \SetRTLapp \SetLTRapp and \SetRTLapp are two argument-less commands to set the direction of the apparatus criticus, either left-to-right or right-to-left.

\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 *New feature* v1.4 \SetSubseparator{⟨*subseparator*⟩} is used to set or change the "subseparator" between succeeding variant readings. By default, no subseparator is set.

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 \ekdsubsep 

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.

\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,

47. See above (d) on page 6.
48. See Sturm (cf. n. 9), 446–9 for details and illustrative examples.

`\SetEndApparatus{.}` will have a period printed at the end of the apparatus as it is customary in some editions.[49]

`\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` *New feature* v1.2 `\SetApparatusLanguage{`⟨*languagename*⟩`}` 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.

`\SetApparatusNoteLanguage` *New feature* v1.3 `\SetApparatusNoteLanguage{`⟨*languagename*⟩`}` 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.

`\SetNegativeApparatus` *New feature* v1.5 `\SetNegativeApparatus` removes the witnesses attached to the lemma text and sets the apparatus criticus in negative form, leaving the reading of lemma text to be understood.

`\SetPositiveApparatus` *New feature* v1.5 `\SetPositiveApparatus` prints the witnesses attached to the lemma text and sets the apparatus criticus in positive form, which is the default.

`\SetTEINegativeApparatus` `\SetTEIPositiveApparatus` *New feature* v1.5 `\SetTEINegativeApparatus` and `\SetTEIPositiveApparatus` do the same as described above for the apparatus criticus converted in `TEI xml`.

`\SetApparatus` **General Command** `\SetApparatus{`⟨*csv list of apparatus settings*⟩`}`

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:—

`direction` `direction=LR|RL` `Default: LR`

The writing direction of the apparatus criticus, either left-to-right (`LR`) or right-to-left (`LR`).

`sep` `sep=`⟨*command* | *chars*⟩ `Default: ]␣`

The separator between lemma texts and variant readings.

`subsep` `subsep=`⟨*command* | *chars*⟩ `Default: not set`

The "subseparator" between succeeding variant readings.

`delim` `delim=`⟨*delimiter*⟩ `Default: not set`

The delimiter between entries in the apparatus criticus. As said above, there is no default delimiter except a simple space.

`bhook` `bhook=`⟨*characters*|*commands*⟩ `Default: empty`

The characters or commands to be appended at the beginning of the apparatus block.

`ehook` `ehook=`⟨*characters*⟩ `Default: empty`

The characters to be appended at the end of the apparatus block.[50]

`rule` `rule=`⟨*command*⟩ `Default: \rule{0.4\columnwidth}{0.4pt}`

As described above, `rule` is used to draw the separating line between the edition text and the apparatus criticus.

`norule` `Default: not set`

`norule` does not accept any value and is used to remove the line.

`lang` `lang=`⟨*languagename*⟩ `Default: not set`

New feature v1.2

49. See also below on page 73 on how to remove superfluous dots.
50. See also n. 49.

lang=<languagename> 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. languagename can be any value accepted by babel or polyglossia.

notelang `notelang=⟨languagename⟩` Default: not set

New feature v1.3 notelang=<languagename> is used as described on the previous page when 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. languagename can be any value accepted by babel or polyglossia.

negative `negative=true|false` Default: false

This named argument does not need a value as it defaults to true if used. negative=true sets the apparatus criticus in print in negative form as described above on the previous page, whereas negative=false can set it back in positive form, which is the default.

TEInegative `TEInegative=true|false` Default: false

This named argument does not need a value as it defaults to true if used. TEINegative=true has the TEI xml apparatus converted in negative form, whereas TEINegative=false can be used to set it back in positive form, which is the default.

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 \footnoterulefalse **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 LaTeX "footnoterule" printed on every page where regular footnotes are 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 35 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:—

```
% preamble:
\DeclareApparatus{testium}
\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
sep
subsep
delim
bhook
ehook
rule
norule
lang
notelang
negative
TEInegative

With regard to layout, any declared layer inherits the default values described above in sect. 6.2 on page 35. 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.

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

```
\SetHooks{
  refnumstyle=\normalfont,
  initialrule=\rule{0.7in}{0.4pt}
}
\DeclareApparatus{testium}[
  sep={] },
  delim=\hskip 1em,
  norule
]
\DeclareApparatus{default}[
  sep={: },
  delim=\hskip 1em,
  rule=\relax
]
```

Rem. 1 The general hook `initialrule` used here (l. 3) is described above on page 34.

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 34. 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^{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. 13.1 on page 70.

7 Inserting Notes in Multiple-Layer Apparatus

As said above in sect. 6.3 on page 37, 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 page 37).

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 testium*.


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 17. 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.[51]

`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`

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.

`labele` `labele=`⟨*label*⟩

`labele` is the unique label to serve as a reference for the point immediately following the lemma text. Contrary to `labelb`, `labele` may be left unspecified if the note be only about

51. See sect. 6.3.1 on page 37 to learn how to declare and lay out new layers of notes.

one word. If the note be about a span, then `labele` must be specified.

`\linelabel` `\linelabel{`⟨*label*⟩`}`

If `labele`=⟨*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, labele=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:—

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. They had a class of nobles, but their tongue knew no word for slave.[52]

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,
```

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

```
<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, labele=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:—

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

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

1 dummy] pseudo

`TEI xml` output:—

```
<p>This is some
<app>
  <lem>
  <anchor xml:id="bnote" />dummy</lem>
  <note type="fontium"
  target="#range(right(bnote),left(enote))">Text of the
  note.</note>
  <rdg>pseudo</rdg>
</app>text to demonstrate how to insert a note in the
argument of the
<emph>lem</emph>command.
<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 Footnotes

8.1 Regular Footnotes

Regular footnotes are inserted by the standard `\footnote` command. When the texts are arranged in parallel columns or on facing pages as described in sect. 5 on page 27, further options are available to specify how the notes should be printed on the pages.

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.[53]

The footmisc package[54] is not compatible with ekdosis. More precisely, footmisc defines a command `\footnotelayout` that is also defined by the paracol package, which ekdosis uses. A simple way around this problem would be to replace `\footnotelayout` in a patched `footmisc-patched.sty` with another name.

`\SetFootnotes` *New feature* v1.5

`\SetFootnotes{`⟨*csv list of footnote settings*⟩`}` can be used either in the preamble or at any point of the document. However, the options for regular footnotes must be set outside the `alignment` environment. This command accepts the following list of `key-value` optional arguments:—

`arrangement` `arrangement=column|page|merge` `Default: not set`

(a) `arrangement=column` has regular footnotes printed at the bottom of the column in which they are called.

(b) `arrangement=page` brings all footnotes together from all columns in a single spanning block at the bottom of all columns on the page.

(c) `arrangement=merge` *merge* 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.

`reset` (no-value argument)

If used, this option reverts the footnotes to the default arrangement which is to have them printed as regular footnotes at the bottom of each column of text. `reset` also reverts footnotes combined into a single paragraph—as described below—to regular footnotes.

8.2 Footnotes Combined Into a Single Paragraph

As already said above, the footmisc package is not compatible with ekdosis. Therefore, it cannot be used to reformat short footnotes so as to combine them all into a single paragraph. This is why ekdosis provides a mechanism of its own to insert such footnotes in a specific layer of the apparatus criticus.

Unlike the three possible values that can be passed to `arrangement` just described, which apply only to regular footnotes inserted in the `alignment` environment, the following options apply to both the `alignment` and `ekdosis` environments and can be set at any point of the document.

It is however not recommended to use `\SetFoonotes` in environments to be transcribed into `TEI xml`. The way to restrict the effect of this command to one of the defined environments is to use `\AtBeginEnvironment` as described in sect. 5.1.1 on page 32.

`\SetFootnotes` (*Continued from above.*)

`paragraph` `paragraph=true|false` `Default: true, initially false`

This named argument does not need a value as it defaults to `true` if used. `paragraph` instructs ekdosis to reformat the notes into a single paragraph, itself inserted as a layer in the apparatus block. (See '`type`' below.)

`type` `type=`⟨*type*⟩ `Initially set as 'default'`

The keyword `type` is used here in the exact same meaning as described above on page 13. '`type`' refers to any layer of critical notes defined by means of `\DeclareApparatus`[55] as fit to receive the footnotes shaped as a single paragraph.

`textfnmark` `textfnmark={`⟨*code*⟩`}` `Default: \textsuperscript{#1}`

53. 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.

54. Frank Mittelbach and Robin Fairbairns, *The Footmisc package* (version 6.0e) [A range of footnote options] (May 26, 2022), https://ctan.org/pkg/footmisc.

55. See above sect. 6.3.1 on page 37.

`textfnmark` takes as value the code used to format the footnote mark that is inserted in the main text. `#1` is the placeholder for the figure that refers to the footnote. For instance, `textfnmark=\textsuperscript{#1}` prints the footnote mark superscripted.

appfnmark `appfmark={`⟨*code*⟩`}` `Default: \textsuperscript{#1}`

`appfnmark` operates on the mark that is inserted in the footnote block the same way as `textfnmark` does on the footnote mark in the main text. As an example, `appfnmark=\hskip 1em\textsuperscript{#1}` first inserts a generous spacing between the items then prints the mark superscripted.

\footnote `\footnote[`⟨*options*⟩`]{`⟨*text*⟩`}` When the mechanism described in this section is set in motion, the `\footnote` command is redefined so as to accept the following list of optional arguments:—

type `type=`⟨*type*⟩ `Default: as defined in \SetFoonotes`

`type` allows to specify any layer of critical notes in which the contents of the footnote is to be printed.

mark `mark=`⟨*mark*⟩ `Default: \thefootnote`

`mark` allows to print any other mark in place of the standard footnote number. When this option is used, the `footnote` counter is not incremented.

\footnotemark \footnotetext `\footnotemark[`⟨*mark*⟩`]` and `\footnotetext[`⟨*options*⟩`]{`⟨*text*⟩`}` are the companions of `\footnote`. The former replaces the definition of and behaves as its standard counterpart while the latter does the same as the `\footnote` command described above and accepts the same optional arguments except that it does not print any mark in the main text.

It is however not advisable to use `\footnotemark` and `\footnotetext` for these commands are not suitable for `TEI xml` export.

The following example illustrates how short footnotes can be combined into one single layer of critical notes:—

```
% Preamble:
\DeclareApparatus{default}
\DeclareApparatus{notes}[bhook=\textbf{Notes:}]
\SetFootnotes{
  paragraph,
  type = notes,
  textfnmark = \textsuperscript{#1},
  appfnmark = \unskip\hskip 1em\textsuperscript{#1}
}

% Document:
\begin{ekdosis}
  (From \cite[\enquote{Dirty Tricks}, \pno~395]{KnuthTeXBook}) And now
  for our next trick,\footnote{First footnote.} let's consider an
  application to short footnotes.\footnote{Second footnote. (Every
    once in a while, a long footnote might occur, just to make things
    difficult.)} The footnotes\footnote{Third footnote.} at the bottom
  of this page \app{\lem{look}\rdg{should look}}
  funny,\footnote{Fourth footnote.}  because most of
  them\footnote[mark=*, type=default]{This note \emph{doesn't count},
    literally. And for some reason, I decided it should sneak into the
    default layer of critical notes.} are quite short.\footnote{Fifth
    footnote. (This is incredibly boring, but it's just an example.)}
  When a document has lots of footnotes,\footnote{Another.} and when
  most of them\footnote{And another.} take only a small part of a
  line,\footnote{Ho hum.}  the output routine\footnote{Umpteenth
    footnote.} ought to reformat them in some more appropriate
```

```
  way.\footnote{Oodles of them.}
\end{ekdosis}
```

PDF output:—

(From Donald E. Knuth, *The TEXBook* (32nd edn., Reading, Mass.: Addison–Wesley, 2013), "Dirty Tricks", p. 395) And now for our next trick,[1] let's consider an application to short footnotes.[2] The footnotes[3] at the bottom of this page look funny,[4] because most of them[*] are quite short.[5] When a document has lots of footnotes,[6] and when most of them[7] take only a small part of a line,[8] the output routine[9] ought to reformat them in some more appropriate way.[10]

---

3 look] should look [*]This note *doesn't count*, literally. And for some reason, I decided it should sneak into the default layer of critical notes.

Notes: [1]First footnote. [2]Second footnote. (Every once in a while, a long footnote might occur, just to make things difficult.) [3]Third footnote. [4]Fourth footnote. [5]Fifth footnote. (This is incredibly boring, but it's just an example.) [6]Another. [7]And another. [8]Ho hum. [9]Umpteenth footnote. [10]Oodles of them.

TEI xml output:—

```
<p>(From
<bibl corresp="#KnuthTeXBook">
  <biblScope>
  <quote>Dirty Tricks</quote>, p. 395</biblScope>
</bibl>) And now for our next trick,
<note place="bottom">First footnote.</note>let's consider
an application to short footnotes.
<note place="bottom">Second footnote. (Every once in a
while, a long footnote might occur, just to make things
difficult.)</note>The footnotes
<note place="bottom">Third footnote.</note>at the bottom of
this page
<app>
  <lem>look</lem>
  <rdg>should look</rdg>
</app>funny,
<note place="bottom">Fourth footnote.</note>because most of
them
<note type="default" place="bottom">This note
<emph>doesn't count</emph>, literally. And for some reason,
I decided it should sneak into the default layer of
critical notes.</note>are quite short.
<note place="bottom">Fifth footnote. (This is incredibly
boring, but it's just an example.)</note>When a document
has lots of footnotes,
<note place="bottom">Another.</note>and when most of them
<note place="bottom">And another.</note>take only a small
part of a line,
<note place="bottom">Ho hum.</note>the output routine
<note place="bottom">Umpteenth footnote.</note>ought to
reformat them in some more appropriate way.
<note place="bottom">Oodles of them.</note></p>
```

▶ Obviously, the mechanism just described only operates on texts equipped with an apparatus criticus. A way to have footnotes shaped as a single paragraph attached to a text that includes no variants, such as a translation arranged on the facing page, is therefore

to format this translation environment as if it were to receive an apparatus criticus, then remove the line numbering from it, like so:—

```
% Preamble:
\SetAlignment{
  texts=edition;translation,
  apparatus=edition;translation
}
\AtBeginEnvironment{translation}{
  \SetLineation{lineation=none}
}
```

9 Poetry

9.1 The Standard `verse` Environment

In order to typeset verse texts or poems, LaTeX 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 TeX 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?\\[2ex]
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.

9.2 The verse Package

New feature v1.2

ekdosis can use the facilities offered by the excellent verse package[56] 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.[57]

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.[58] In this case, ekdosis automatically uses the code provided by this class without the need to set the global option `poetry=verse`.

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}[`⟨*options*⟩`] ... \end{ekdverse}`

This environment is used to hold verse lines as described above and may receive an optional

56. 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.
57. See on page 58 for details.
58. 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.

argument in which the following "name=value" arguments are accepted:—

width `width`=⟨*length*⟩ 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 LaTeX 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`=⟨*type*⟩ 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 hiearchichally. Within this environment, `ekdstanza` must be used instead to encode stanzas as described below on the following page.

As an example, the first five lines of Homer's *Odyssey* could be encoded like so:[59]—

```
\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 πολλὰ \\
      πλάγχθη, ἐπεὶ Τροίης ἱερὸν πτολίεθρον ἔπερσεν· \\
      πολλῶν δ' ἀνθρώπων ἴδεν ἄστεα καὶ
                    \app{\lem{νόον}
                         \rdg[resp=Zen]{νόμον}
                         \note{Cf. Schol.}} ἔγνω, \\
      πολλὰ δ' ὅ γ' ἐν πόντῳ πάθεν ἄλγεα ὃν κατὰ θυμόν, \\
      ἀρνύμενος ἥν τε ψυχὴν καὶ νόστον ἑταίρων. \\
    \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:—

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

Ἄνδρα μοι ἔννεπε, Μοῦσα, πολύτροπον, ὃς μάλα πολλὰ
πλάγχθη, ἐπεὶ Τροίης ἱερὸν πτολίεθρον ἔπερσεν·
πολλῶν δ᾽ ἀνθρώπων ἴδεν ἄστεα καὶ νόον ἔγνω,
πολλὰ δ᾽ ὅ γ᾽ ἐν πόντῳ πάθεν ἄλγεα ὃν κατὰ θυμόν,
ἀρνύμενος ἥν τε ψυχὴν καὶ νόστον ἑταίρων.

3 νόον] νόμον Zen. Cf. Schol.

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.

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' ὅ γ' ἐν πόντῳ πάθεν ἄλγεα ὃν κατὰ θυμόν,</l>
    <l>ἀρνύμενος ἥν τε ψυχὴν καὶ νόστον ἑταίρων.</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 LaTeX `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}[`⟨*options*⟩`] ... \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=`⟨*type*⟩ `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,[60] the indentation pattern consists of an array of digits, d_1 to d_n, where the n^{th} line is indented by d_n times the amount of `\vgap` described above on page 47.

60. Wilson and Robertson (cf. n. 56), 6.

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

```
\begin{ekdverse}[type={overall type}]
  \identpattern{digits}
  \begin{patverse}
    \begin{ekdstanza}[type={stanza 1 type}]
      line 1 \\
      line 2 \\
      [...]
      final line \\!
    \end{ekdstanza}
  \end{patverse}
\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:—

```
\begin{ekdverse}[type={overall type}]
  \begin{ekdstanza}[type={stanza 1 type}]
    line 1 \\
    line 2 \\
    \vin indented line 3 \\
    [...]
    final line \\!
  \end{ekdstanza}
\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.[61] 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.[62] However, this typographical refinement has been retained for the sonnet only. For the sake of clarity, 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

```
\junicode % Use the Junicode font with 'hist' feature enabled for
          % long-s
% MacDonald's style for numbers and entries in the apparatus
% criticus:
\SetHooks{
  refnumstyle=\normalfont,
  postrefnum=.~,
  lemmastyle=\bfseries,
  readingstyle=\itshape,
  familysep={, }
}
% The lines are to be centered horizontally:
```

61. 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.

62. 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.

```
\settowidth{\versewidth}{Feed'st thy lights flame with selfe
substantiall fewell,}
% Format of the outermost <div> element:
\NewDocumentEnvironment{ekdcenter}{}{\par\centering}{\nobreak\par}
\FormatDiv{1}{\begin{ekdcenter}}{\end{ekdcenter}}
\begin{ekdosis}
  \ekddiv{type=sonnets, n=1, head=1}
  \begin{ekdverse}[type=sonnet, width=\versewidth]
    \indentpattern{00000000000011}
    \begin{patverse}
      \begin{ekdstanza}[type=quatrain]
        \ekdlettrine{F}{r}om fairest creatures we desire
        increase,\\
        That thereby beauties \emph{Rose}
        \app{
          \lem{might}
          \rdg[source={Gildon1710, Sewell1725, Ewing1771}]{may}
        } neuer die,\\
        But as the riper should by time
        \app{
          \lem{decease}
          \rdg[source=Hudson1856]{decrease}
        },\\
        His tender heire might beare his memory:\\
      \end{ekdstanza}
      \begin{ekdstanza}[type=quatrain]
        But thou contracted to thine owne bright eyes,\\
        Feed'st thy
        \app{
          \lem{lights}
          \rdg[source={Butler1899, Walsh1908}]{life's}
        } flame with
        \app{
          \lem{selfe substantiall}
          \rdg[source=Gildon1714,
          alt={\textnormal{Hyphened by}}]{selfe-substantiall}
          \note{etc.}} fewell,\\
        Making a famine where aboundance lies,\\
        Thy selfe thy foe,to thy sweet selfe too cruell:\\
      \end{ekdstanza}
      \begin{ekdstanza}[type=quatrain]
        Thou that art now the worlds fresh ornament,\\
        And
        \app{
          \lem{only}
          \rdg[resp=God, type=conjecture]{early}
          \note{conj.}
        } herauld to the gaudy spring,\\
        Within thine owne bud buriest thy content,\\
        And tender
        \app{
          \lem{chorle}
          \rdg[source=Gildon1710]{churl}
          \note{etc.}
        } makst wast in niggarding:\\
```

```
      \end{ekdstanza}
      \begin{ekdstanza}[type=couplet]
        Pitty the world,or else this glutton be,\\
        To eate the worlds due,\app{
          \lem{by the}
          \rdg[resp=Stee, type=conjecture]{be thy}
          \note{conj.}
          \rdg[resp=God, type=conjecture]{by thy}
          \note{conj.}
        } graue
        \app{
          \lem{and}
          \rdg[resp=God, type=conjecture]{as}
          \note{conj.}
        } thee.\\!
      \end{ekdstanza}
    \end{patverse}
  \end{ekdverse}
\end{ekdosis}
```

Rem. 1 `Gildon1710`, `Gildon1714`, `Sewell1725`, `Ewing1771`, `Hudson1856`, `Butler1899` and `Walsh1908` have been declared as sources.[63] `God` and `Stee`, resp. Godwin and Steevens, have been declared as scholars.[64]

Rem. 2 `\ekdlettrine` (l. 23) is a specific command for the lettrine package does not work in list environments. `\TeXtoTEIPat` has been used as described below on page 80 to instruct ekdosis to convert this command into an acceptable `TEI` equivalent. The definition of `\ekdlettrine` 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{\ekdlettrine}{mm}{%
\adjustbox{valign=t,raise=-0.75ex}{\Huge #1}\textsc{#2}%
}
\TeXtoTEIPat{\ekdlettrine {#1}{#2}}{<hi rend="smallcaps">#1#2</hi>}
```

PDF output:—

I

From faireſt creatures we deſire increaſe, 1
That thereby beauties *Roſe* 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 aboundance 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 burieſt thy content, 11
And tender chorle makſt waſt in niggarding: 12

2. **might**] *may* G, S, E 3. **decease**] *decrease* Hu² 6. **lights**] *life's* But, Wa **selfe substantiall**] Hyphened by G² etc. 10. **only**] *early* Godwin conj. 12. **chorle**] *churl* G etc.

63. See above on page 10 and below sect. 14.7 on page 83.
64. See above on page 11.

Pitty the world,or elſe this glutton be, 13
To eate the worlds due,by the graue and thee. 14

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>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>
```

9.3 Arabic Poetry

New feature 1.5 A short introduction to editing Arabic texts with ekdosis and arabluatex is provided below, sect. 11.2 on page 61, "Using arabluatex". With regard to the technique of typesetting Arabic poetry, the reader is invited to refer to the relevant section of the documentation of arabluatex.[65]

ekdosis *must be loaded* with the `poetry=verse` option as described on page 7. Then, the `arabverse` environment is to be used instead of `ekdverse` to typeset the lines of Arabic poetry.

`\bayt+` As of v1.21 of arabluatex, the `\bayt{`⟨*ṣadr*⟩`}[`⟨*tadwīr*⟩`]{`⟨*ʿağuz*⟩`}` command it provides accepts a `+` optional argument that is to be used to let critical notes be inserted in lines of poetry, like so:—

```
% Preamble:
\usepackage[poetry=verse]{ekdosis}

% Document:
\begin{ekdosis}
  \begin{arabverse}[<options>]
    \bayt{<ṣadr>}[<tadwīr>]{<ᶜağuz>}\\ % no critical notes
    [...]
    \bayt+{\app{ % example of \bayt+ with critical notes
        \lem{Sample}
        \rdg{Example}
      } Arabic text}{Sample Arabic text}\\
    [...]
    \bayt{<ṣadr>}[<tadwīr>]{<ᶜağuz>}\\! % last line (no critical notes)
```

65. Robert Alessi, *The Arabluatex package* (version 1.20) [ArabTeX for LuaLaTeX] (Mar. 23, 2020), http://ctan.org/pkg/arabluatex, see "Arabic Poetry".

```
  \end{arabverse}
\end{ekdosis}
```

An example of how one could insert variant readings in Imruʾ al-Qays' *Muʿallaqah* (ll. 26–30) follows:—

Listing 9: Poetry: Imruʾ al-Qays' *Muʿallaqah*

```
1  \begin{alignment}[tcols=1,
2                    texts=specimen,
3                    apparatus=specimen]
4    \resetlinenumber\resetvlinenumber[26]
5    \begin{specimen}
6      \begin{arabverse}[mode=fullvoc, width=0.25\columnwidth]
7        \bayt+{\app{
8          \lem{ta^gAwaztu 'a.hrAsaN}
9          \rdg{ta_ha.t.taytu 'abwAbaN}
10         \rdg[subsep={:~}]{ta_ha.t.taytu 'ahwAlaN}
11       } 'ilay-hA wa-ma`^saraN}{`alayya
12      .hirA.saN law \app{
13        \lem[nonum]{yusirrUna}
14        \rdg{yu^sirrUna}
15      } maqtalI}\\
16     \bayt{'i_dA mA 'l-_turayyA fI 'l-samA'i ta`arra.dat}{ta`arru.da
17       'a_tnA'i 'l-wi^sA.hi 'l-mufa.s.sali}\\
18     \bayt{fa-^gi'tu wa-qad na.d.dat li-nawmiN _tayAna-hA}{lad_A
19       'l-sitri 'illA libsaTa 'l-mutafa.d.dili}\\
20     \bayt{fa-qAlat yamInau 'l-l_ahi mA la-ka .hIlaTuN}{wa-mA 'in
21     'ar_A `an-ka 'l-.gawAyaTa tan^galI}\\
22     \bayt+{_hara^gtu bi-hA \app{
23         \lem{tam^sI}
24         \rdg{'am^sI}
25       } ta^gurru wa-rA'a-nA}{`al_A 'a_trinA
26         'a_dyAla mir.tiN mura.h.hali}\\!
27     \end{arabverse}
28   \end{specimen}
29 \end{alignment}
```

Rem. `\bayt+` has been applied twice, at lines 7 and 22. In the first instance (ll. 7–15), variant readings have been inserted in both hemistichs. As the hemistichs are technically enclosed in separate TEX groups, ekdosis naturally recalls in the apparatus criticus the line number corresponding to the first entry of each one of the two groups. An easy way to get around this is to apply the `nonum` optional argument to the first `\lem` command used in the second hemistich, as has been done here on line 13.

PDF output:—

26 تَجَاوَزْتُ أَحْرَاسًا إِلَيْهَا وَمَعْشَرًا — عَلَيَّ حِرَاصًا لَوْ يُسِرُّونَ مَقْتَلِي

27 إِذَا مَا الثُّرَيَّا فِي السَّمَاءِ تَعَرَّضَتْ — تَعَرُّضَ أَثْنَاءِ الْوِشَاحِ الْمُفَصَّلِ

28 فَجِئْتُ وَقَدْ نَضَّتْ لِنَوْمٍ ثِيَابَهَا — لَدَى السِّتْرِ إِلَّا لِبْسَةَ الْمُتَفَضِّلِ

29 فَقَالَتْ يَمِينُ اللهِ مَا لَكَ حِيلَةٌ — وَمَا إِنْ أَرَى عَنْكَ الْغَوَايَةَ تَنْجَلِي

30 خَرَجْتُ بِهَا تَمْشِي تَجُرُّ وَرَاءَنَا — عَلَى أَثَرِنَا أَذْيَالَ مِرْطٍ مُرَحَّلِ

26 تَجَاوَزْتُ أَحْرَاسًا] تَخَطَّيْتُ أَبْوَابًا : تَخَطَّيْتُ أَهْوَالًا يُسِرُّونَ] يُشِرُّونَ **30** تَمْشِي] أَمْشِي

TEI xml output:—

```
<lg xml:lang="ar-Latn" type="transliterated"
subtype="arabtex">
  <l>
    <seg type="hemistich">
    <app>
      <lem>ta^gAwaztu 'a.hrAsaN</lem>
      <rdg>ta_ha.t.taytu 'abwAbaN</rdg>
      <rdg>ta_ha.t.taytu 'ahwAlaN</rdg>
    </app>'ilay-hA wa-ma`^saraN</seg>
    <seg type="hemistich">`alayya .hirA.saN law
    <app>
      <lem>yusirrUna</lem>
      <rdg>yu^sirrUna</rdg>
    </app>maqtalI</seg>
  </l>
  <l>
    <seg type="hemistich">'i_dA mA 'l-_turayyA fI 'l-samA'i
    ta`arra.dat</seg>
    <seg type="hemistich">ta`arru.da 'a_tnA'i 'l-wi^sA.hi
    'l-mufa.s.sali</seg>
  </l>
  <l>
    <seg type="hemistich">fa-^gi'tu wa-qad na.d.dat
    li-nawmiN _tayAna-hA</seg>
    <seg type="hemistich">lad_A 'l-sitri 'illA libsaTa
    'l-mutafa.d.dili</seg>
  </l>
  <l>
    <seg type="hemistich">fa-qAlat yamInau 'l-l_ahi mA
    la-ka .hIlaTuN</seg>
    <seg type="hemistich">wa-mA 'in 'ar_A `an-ka
    'l-.gawAyaTa tan^galI</seg>
  </l>
  <l>
    <seg type="hemistich">_hara^gtu bi-hA
    <app>
      <lem>tam^sI</lem>
      <rdg>'am^sI</rdg>
    </app>ta^gurru wa-rA'a-nA</seg>
    <seg type="hemistich">`al_A 'a_trinA 'a_dyAla mir.tiN
    mura.h.hali</seg>
  </l>
</lg>
```

10 Lineation Settings

ekdosis uses lineno internally for line numbering.[66] But it must be noted that ekdosis strictly prohibits the "pagewise" mode of operation that is provided by lineno. As a result of this hinderance, all "margin switching" functions of lineno are disabled within the environments that are specific to ekdosis, viz. `ekdosis` and `alignment`.

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

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` `\outerlinenumbers` `\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.

Limiting the Number of Lines per Page

`maxlines` `maxlines=`*n* (where *n* is an integer ≥ 1) `Default: not set`

New feature v1.5 `maxlines=`*n* instructs ekdosis to break the pages of numbered text every *n* lines. This option is very useful for building editions equipped with long and complex apparatus criticus. The rationale is to start with a number of lines that will allow all pages to pass just after a few runs of LuaLaTeX, even at the cost of showing blanks between the edition text and the apparatus criticus. Adjusting further the number of lines will then leave fewer blanks on display. Combined with the `fitapp` global option[67] or with `maxentries`,[68] `maxlines` can achieve excellent results. For more details on this technique, see below sect. 13.1, "The Oscillating Problem" on page 70.

`nomaxlines` `Default: not set`

New feature v1.5 This no-value option unsets any limit previously set by `maxlines` or `\setmaxlines`

`\setmaxlines` `\setmaxlines{`⟨*n*⟩`}` (where ⟨*n*⟩ ≥ 1) has the same effect as the `maxlines` option just described. This command can be used either in the preamble or at any point of the document.

▶ The `maxlines` option operates globally, on any pages or columns of text that are set to receive at least one layer of apparatus criticus. The way of applying the limit to only one out of several edition texts is therefore to append `\setmaxlines` as a hook to the environment corresponding to this edition text as described above on page 32, like so:—

67. See above on page 6.
68. See above on page 38.

```
\AtBeginEnvironment{latin}{\setmaxlines{<n>}}
```

\nomaxlines `\nomaxlines` is an argument-less command that operates as the `nomaxlines` option just described. This command can be used either in the preamble or at any point of the document.

Options Specific to the `poetry=verse` Global Setting[69]

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^{th} 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 `\\+`[70] 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.

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. 9.2 on page 47, the commands dedicated to line numbering that are provided by the verse package can be used.

\vmodulolinenumbers `\vmodulolinenumbers[`⟨*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*⟩`}`.

69. See above sect. 9.2 on page 47.
70. See above (d) to (e) on page 47 for more information.

\linelabel
\ref

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
}
```

11 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 35, 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,[71] 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. 14 on page 74 including the example provided on page 80.

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 page 36 for single-layer apparatus and on page 38 for multiple-layer apparatus.

11.1 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.[72]

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

Listing 10: Multilingual editions with babel only

71. 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.

72. 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.

```
1  \usepackage{fontspec}
2
3  \usepackage[greek.ancient,english]{babel}
4  \babelprovide[onchar=fonts]{arabic}
5
6  \babelfont{rm}{Old Standard}
7  \babelfont[greek]{rm}[RawFeature={+ss05;+ss06}]{Old Standard}
8  \babelfont[*arabic]{rm}{Amiri}
9
10 \babeltags{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 `\babelfont` 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 `\babelfont` 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{ancient greek}` ... `\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.

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` `\setLR`

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 form 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 11: 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 أَبِي كَانَ مِنَ \app{
      \lem{ٱلْمُقَاتِلَةِ}
      \rdg{ٱلْمُقَاتِلِينَ}
      } وَ كَانَتْ أُمِّي مِنْ عُظَمَآءِ بُيُوتِ ٱلزَّمَازِمَةِ۔
  \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.[73]

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

```
1
```

إِنَّ أَبِي كَانَ مِنَ ٱلْمُقَاتِلَةِ وَكَانَتْ أُمِّي مِنْ عُظَمَآءِ بُيُوتِ ٱلزَّمَازِمَةِ.

1 ٱلْمُقَاتِلِينَ [ٱلْمُقَاتِلَةِ

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

```
1
```

إِنَّ أَبِي كَانَ مِنَ ٱلْمُقَاتِلَةِ وَكَانَتْ أُمِّي مِنْ عُظَمَآءِ بُيُوتِ ٱلزَّمَازِمَةِ.

1 ٱلْمُقَاتِلَةِ] ٱلْمُقَاتِلِينَ

11.2 Using arabluatex

arabluatex is a LuaLaTEX package that provides commands and environments which return Arabic writing from an ASCII transliteration (either ArabTEX or Buckwalter scheme).[74] It is particularly well-suited for complex documents such as critical editions where a lot of commands intertwine with Arabic writing. arabluatex can output Unicode Arabic in the same modes as arabtex[75] or in different accepted standards of romanization. It is also able to produce a duplicate of the original `.tex` source file in which all `arabtex` or `buckwalter`

73. See above sect. 6.2 on page 35 (single-layer apparatus criticus) and sect. 6.3 on page 37 (multiple-layer apparatus criticus).
74. Alessi, *The Arabluatex package* (cf. n. 65).
75. 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.

strings are replaced with Unicode equivalents, either in Arabic script or in any accepted standard of transliteration.[76]

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

Listing 12: ekdosis and arabluatex

```
% Preamble:
% load ekdosis and ask for TEI xml output:
\usepackage[teiexport]{ekdosis}
% load arabluatex and request a LaTeX output with Unicode Arabic:
\usepackage[export,fullvoc]{arabluatex}

% 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}
```

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

1 إِنَّ أَبِي كَانَ مِنَ ٱلْمُقَاتِلَةِ وَكَانَتْ أُمِّي مِنْ عُظَمَآءِ بُيُوتِ ٱلزَّمَازِمَةِ.

1 ٱلْمُقَاتِلِينَ]ٱلْمُقَاتِلَةِ

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 12, 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
<app>
  <lem>ٱلْمُقَاتِلَةِ</lem>
```

76. Alessi, *The Arabluatex package* (cf. n. 65), "Exporting Unicode Arabic to an External File."

```
  <rdg>ٱلْمُقَاتِلِينَ</rdg>
</app>وَكَانَتْ أُمِّي مِنْ عُظَمَآءِ بُيُوتِ
ٱلزَّمَازِمَةِ.</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.[77]

12 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 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 LaTeX source files with Lua functions which involve complex string matching and recursions was required, as LaTeX '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.[78] As a result, ekdosis will disregard whichever one is not selected.

12.1 LaTeX Standard Divisions

ekdosis can use the LaTeX 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 LaTeX textual divisions are converted into numbered `<div`$_n$`>` 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. 12.2 on the next page.

Once `divs` has been set to `latex`, ekdosis converts `\book`, `\part`, `\chapter`, `\section`, `\subsection` and `\subsubsection` into corresponding `TEI` 'numbered' `<div`$_n$`>` 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 LaTeX 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.

77. Alessi, *The Arabluatex package* (cf. n. 65).
78. See https://tei-c.org/release/doc/tei-p5-doc/en/html/DS.html#DSDIV.

Inserting Variants in Headings Variant readings can be inserted in headings. In this case, the optional argument of the LaTeX sectional command must naturally be used to prevent variants from going into headers, footers or the table of contents, like so:[79] —

```
% Preamble:
\usepackage[teiexport=tidy, divs=latex]{ekdosis}
\MkBodyDivs{chapter}{section}{}{}{}{}

% Document:
\chapter[Ἱπποκράτους ἐπιδημιῶν βιβλίον δεύτερον]{Ἱπποκράτους ἐπιδημιῶν
  \app{
    \lem[wit={I,R,H}]{βιβλίον δεύτερον}
    \rdg[wit=V]{λόγος β’}}}

\section[Τμῆμα πρῶτον]{
  \app{
    \lem[resp=egomute, type=emendation, nosep,
     post=suppleui]{\supplied{Τμῆμα πρῶτον}}
   }}
Ἄνθρακες θερινοὶ ἐν Κραννῶνι· [...]
```

TEI xml output:—

```
<div1 type="chapter">
  <head>Ἱπποκράτους 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">Τμῆμα
         πρῶτον</supplied></lem>
      </app>
    </head>
    <p>Ἄνθρακες θερινοὶ ἐν Κραννῶνι· [...]</p>
  </div2>
</div1>
```

12.2 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

79. On the use of `egomute` (l. 13), see above Rem. 2 on page 22.

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, a numeric value `n` is needed to indicate the depth of the division within the hierarchy, the largest being `1` and the smallest being `9`.

It must be noted that from one division level to the next, the upward progression must be continuous. For example, moving from `n=2` to `n=3` is allowed, whereas `n=2` to `n=4` will not produce a valid `xml` code.

`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. 12.3 on page 67.

`\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. 12.1 on page 63, here follows how the first three hierarchical levels could be formatted as un-numbered divisions:—

Listing 13: Divisions of the body text

```
% Preamble:
\FormatDiv{1}{\begin{center}\Large}{\end{center}}
\FormatDiv{2}{\begin{center}\large}{\end{center}}
\FormatDiv{3}{\bfseries}{.}

% Document:
\begin{ekdosis}
  \ekddiv{
    head={Ἱπποκράτους ἐπιδημιῶν
      \app{
        \lem[wit={I,R,H}]{βιβλίον δεύτερον}
        \rdg[wit=V]{λόγος β'}}},
```

```
    type=book,
    depth=1,
    n=II
  }

  \ekddiv{
    head={\app{
        \lem[resp=egomute, post=suppleui,
        type=emendation]{\supplied{Τμῆμα πρῶτον}}
        \rdg[wit=codd, alt=om.]{}}},
    type=section,
    depth=2,
    n=II.1
  }

  \ekddiv{head=1, type=paragraph, depth=3, n=II.1.1}
  Ἄνθρακες θερινοὶ ἐν Κραννῶνι· [...]
\end{ekdosis}
```

PDF output:—

Ἱπποκράτους ἐπιδημιῶν βιβλίον δεύτερον

⟨Τμῆμα πρῶτον⟩

1. Ἄνθρακες θερινοὶ ἐν Κραννῶνι· [...]

1 βιβλίον δεύτερον IRH] λόγος β’ V **2** Τμῆμα πρῶτον suppleui] om. codd.

TEI xml output:—

```
<div xml:id="div-hippocrates_1" xml:lang="grc">
  <div type="book" n="II">
    <head>Ἱπποκράτους 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>Τμῆμα 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>
```

12.3 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. 12.2 on page 64.

`\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:

`\thepage`	Hippocratis	

Odd pages:

	Epidemiarum liber II, `\ekdmark`	`\thepage`

To return to the example provided by listing 13 on page 65, 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:[80]—

```
% 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:—

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

```
% 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}
```

"Paired" facing pages As already described on page 30—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` *New feature* v1.5

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}
```

New feature v1.5

"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 LaTeX.

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:—

```
% Preamble:
\usepackage{titleps}
\newpagestyle{edition}{
  \sethead[\setpairedpage\ekdprintmark{HEL}{\thepage}]
          [\ekdprintmark{HEC}{Hippocratis}]
          []
          {}
          {\setpairedpage\ekdprintmark{HOC}{Epidemiarum liber II,\
              \ekdmark}}
          {\ekdprintmark{HOR}{\thepage}}\
}
% Apply the page style:
\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 14, l. 16.

A schematic illustration of this technique follows:—

Listing 14: "Mirrored" paired facing pages

```
\mainmatter
\part{Edition text}
\cleartoevenpage     % this needs the 'nextpage' package.
\pagestyle{edition}  % apply the style where 'mirrored' paired facing
                     % pages are defined as illustrated above.
\begin{alignment}
  \begin{edition}
    ...
  \end{edition}
  \begin{translation}
    ...
  \end{translation}
\end{alignment}
\backmatter          % or \cleartooddpage provided by the 'nextpage'
                     % package.
\resetpagenumber     % this command is always required.
\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 LaTeX 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.

13 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 LuaLaTeX to be run three more times to get everything to the right place with the right numbers.

13.1 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 LuaLaTeX 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.

The 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.

Limiting the Number of Lines per Page One way to avoid this inconvenience is to use the `maxlines` option of `\SetLineation` as described above on page 57. Depending on the abundance of critical footnotes to be printed, the editor may start with a number that will allow most, if not all, pages to pass.

Furthermore, this technique can be combined with the `fitapp` global option described below on page 72. Once `maxlines` has been applied, the fgruler package[81] can be used to gauge the respective heights of the edition text and the apparatus criticus like so:—

```
% Preamble
\usepackage[type=lowerleftT]{fgruler}
```

Based on the height corresponding to the last line of the edition text limited by the value of `maxlines`, the height from which the apparatus criticus block should stop growing and the characters should be scaled down to allow for more entries can be estimated.[82]

81. Tibor Tómács, *The Fgruler package* (version 1.5) [Draw rulers on the foreground or in the text] (June 25, 2022), https://ctan.org/pkg/fgruler.

82. To learn how the maximum height of the apparatus criticus can be set, see on page 35. The `fitapp` global option is described on page 6.

This ensures that any contentious entries are included in the apparatus criticus when the last line of text has been reached.

In addition to the `maxlines` option of `\SetLineation` and the `\setmaxlines` command,[83] the following commands are provided:—

`\localmaxlines` `\localmaxlines{⟨n⟩}`, where $\langle n \rangle \geq 1$, can be used in the edition text to adjust the number of lines on a given page. Of course, this command must be issued before the line number corresponding to `maxlines` is reached.

▶ `\localmaxlines{0}` can therefore be used to remove the limit set by `maxlines` or `\setmaxlines` on a given page.

`\addtomaxlines` Unlike `\localmaxlines`, `\addtomaxlines{⟨n⟩}` takes as argument the number of lines one wishes to add or substract from the number that has been set by `maxlines` or `\setmaxlines`. As a result, $\langle n \rangle$ can be a positive or negative integer.

`\nomaxlines` `\nomaxlines` is an argument-less command that unsets any limit previously set by `maxlines` or `\setmaxlines`.

`\ekdpb` `\ekdpb*` *New feature* v1.2

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

One other way 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.[84]

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 34 and 38. 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.

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` *New feature* v1.1

Adding and Removing Entries `\addentries[⟨layer⟩]{⟨n⟩}`

If `maxentries` be set for a given layer of critical notes, `\addentries[⟨layer⟩]{⟨n⟩}`, where $\langle n \rangle$ is an integer, can be used to add $\langle n \rangle$ to—or remove it from if $\langle n \rangle$ be negative—the

83. See above on page 57.
84. This requires the `showpagebreaks` option to be set to `true` as described above on page 7.

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.[85] However, it is advisable to use `fitapp` conjointly with `maxlines` or `maxentries` to prevent the number of allowed entries from being too high, which would result in the characters being too small or even illegible.

New feature v1.5 As already seen on page 35, once the global option `layout=fitapp` has been set,[86] the default maximum height of the apparatus criticus block is `0.5\textheight`, which can be changed by assigning a new length to the `appheight` option of `\SetHooks`. To more finely adjust this height on given pages, ekdosis provides additional commands to be used in the edition text:—

`\localappheight` `\localappheight{`⟨*dimen*⟩`}` can be used to change locally the height up to which the apparatus block is allowed to grow. ⟨*dimen*⟩ must be a number followed by a unit length.

This command operates only on the apparatus block that follows it. Therefore, it must be issued *before* the first entry of the apparatus block on which it is intended to operate, either on the current page or in the last lines of the preceding page.

`\addtoappheight` As the name suggests, in contrast to `\localappheight`, `\addtoappheight{`⟨*dimen*⟩`}` is used to increase or decrease locally the height of the apparatus typeblock. ⟨*dimen*⟩ must be a number followed by a unit length. Just as `\localappheight`, this command operates only on the apparatus block that follows it.

13.2 Using emacs

ekdosis includes an AUCTEX style file that can be used to facilitate the insertion of the many commands it provides.

Installation In what follows, it is assumed that both the emacs text editor[87] and AUCTEX[88] have already been installed.

ekdosis Stable (a) Download `ekdosis.el` from CTAN: https://ctan.org/tex-archive/macros/luatex/latex/ekdosis

(b) Copy this file to `$HOME/.emacs.d/auctex/style/`[89] where `$HOME` stands for the directory of the current user.

ekdosis Development Version (a) Download `ekdosis.el` from either `git.robertalessi.net` (http://git.robertalessi.net/ekdosis/plain/ekdosis.el) or

85. That is, cases that arise *after* the predefined height has been reached.
86. See above on page 6.
87. https://www.gnu.org/software/emacs
88. https://www.gnu.org/software/auctex/download.html
89. This directory must be created if need be.

the sourcehut git repository (https://git.sr.ht/~ralessi/ekdosis/blob/master/ekdosis.el).

(b) Copy this file to $HOME/.emacs.d/auctex/style/[89] where $HOME stands for the directory of the current user.

If desired, ekdosis.el can be compiled like so:—

```
emacs --batch -f batch-byte-compile ekdosis.el &>/dev/null
```

This will produce ekdosis.elc which can be copied to the same directory as ekdosis.el.

Code Folding The variant readings and the critical notes can grow in number to a point where they may clutter the source text. As a result, the edition text can become illegible. One way around this difficulty is to use the emacs editor with ekdosis.el, the AUCTEX style file that is provided with ekdosis, to fold the code so that only the edition text, exclusive of variants and notes, is displayed on the screen.

In order to hide all foldable items, TeX-fold-mode must first be activated, like so: C-c C-o C-f.[90] Then C-c C-o C-b can be used to fold the code.[91]

\App `\App[type=⟨type⟩]{⟨lemma text⟩}{⟨readings and notes⟩}`

New feature v1.5 \App allows for much more flexible code folding where notes and variants are hidden to let only the base text appear on the screen. \App is strictly equivalent to \app, except that the apparatus entries are meant to be distributed in two different arguments, like so:—

```
1 I saw my friend \App{\lem{Peter}}{\rdg{John}} yesterday.
2 or:
3 I saw my friend \App{\lem{Peter}}{
4     \rdg{John}
5   } yesterday.
```

Rem. 1 As can be seen, the first argument of \App is meant to receive the lemma text while \rdg, \note and the like are inserted in the second one.

Rem. 2 As the second argument of \App is the foldable item, a good practice is to keep the lemma text on the same line as the opening brace (l. 3) and to write the continuation of the text just after the closing brace (l. 5).

Code folding, once applied, results in a clean source text with no clutter as follows:—

```
1 I saw my friend Peter yesterday.
```

As there is no point in using this command for anything other than this specific purpose, \app is used in the examples throughout this document.

13.3 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 35, 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.[92] However, if the last word of the

90. Menu sequence: LaTeX » Show/Hide » Fold Mode
91. Menu sequence: LaTeX » Show/Hide » Hide All in Current Buffer
92. See above on pages 35–37.

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
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 LaTeX 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.

14 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 LaTeX source file and its corresponding TEI xml output are printed in full below in sect. 17 on page 91. 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 LaTeX (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).[93]

14.1 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,[94] then places each of the corresponding TEI xml outputs within distinct 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 91, ll. 85–101) which is declared at l. 21. Then, the corresponding xml output is found within a 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.

14.2 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*.[95] 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 LaTeX packages that are part of TeXLive 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 TeX and LaTeX quite exceed what can be afforded by TEI xml. On another hand, many LaTeX 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 LaTeX document into TEI xml can be quite an intricate business. In many cases, however, LaTeX strings are found within environments or groups that are easy to convert into TEI equivalents: unless instructed otherwise, whether such groups

93. The PDF output is available as a separate file.
94. See above sect. 5.1 on page 30.
95. 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: https://www.teibyexample.org/exist/tools/TBEvalidator.htm.

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 LaTeX does not place everything into groups or environments. To take here but a few examples, sectional divisions are marked in LaTeX 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:—

```
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 LaTeX 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`.[96] 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,

96. See above sect. 5.1 on page 30.

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. 14.4 on page 79. A typical use case of autopar=false is provided below in sect. 14.6 on page 82.

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.

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.

14.3 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. 12.1 on page 63 and most of the commands provided by the arabluatex and icite[97] packages. Standard citation commands are also supported as will be described below in sect. 14.8 on page 86:—

LaTeX command	TEI xml element
\textsuperscript{}	<hi rend="sup"></hi>
\textsubscript{}	<hi rend="sub"></hi>
\textbf{}	<hi rend="bold"></hi>
\textit{}	<hi rend="italic"></hi>
\textsc{}	<hi rend="smallcaps"></hi>
\textsf{}	<hi rend="sf"></hi>
\footnote{}	<note place="bottom"></note>
\marginpar{}	<note place="margin"></note>
\enquote⟨*⟩{}	<quote></quote>
\label{label}	<anchor xml:id="label"/>
\linelabel{label}	<anchor xml:id="label"/>
\ref{label}	<ptr ="#label"/>

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

LaTeX command	TEI xml element
\pageref{label}	<ptr ="#label"/>
\vref{label}	<ptr ="#label"/>
\vpageref{label}	<ptr ="#label"/>
\pagebreak⟨[⟨1-4⟩]⟩	no output
\mbox{⟨*text*⟩}	⟨*text*⟩
From the marginnote[98] package:	
\marginnote{}	<note place="margin"></note>
From the extdash[99] package:	
\--- or \===	—
\-- or \==	–
\-/ or \=/	-

As for environments:—

LaTeX environment	TEI xml element
flushright	
flushleft	
center	
quotation	<quote></quote>
quoting	<quote></quote>
verse	<lg></lg>

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 LaTeX equivalent, all is needed is to define an inoperative LaTeX 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>
```

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

99. 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.

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.

14.4 Processing New Commands or Environments

The following three commands are provided to instruct ekdosis how to convert unknown or unusual (LA)TEX commands or environments into `TEI xml` equivalents.

`\TeXtoTEI` `\TeXtoTEI{`⟨*csname*⟩`}{`⟨*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:[100] —

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

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

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

100. 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. 14.7 on page 83 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.

```
% 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:—

```
<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*

\EnvtoTEI⟨*⟩{⟨*env name*⟩}{⟨*TEI element*⟩}[⟨*TEI attribute(s)*⟩]

\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, , <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}
    περὶ πολλοῦ ἂν ποιησαίμην, ὦ ἄνδρες, τὸ τοιούτους ὑμᾶς ἐμοὶ
    δικαστὰς περὶ τούτου τοῦ πράγματος γενέσθαι, οἵοίπερ ἂν ὑμῖν
    αὐτοῖς εἶητε τοιαῦτα πεπονθότες...
  \end{ancientgreek}
\end{ekdosis}
```

TEI xml output:—

```
<p xml:lang="grc">περὶ πολλοῦ ἂν ποιησαίμην, ὦ ἄνδρες, τὸ
τοιούτους ὑμᾶς ἐμοὶ δικαστὰς περὶ τούτου τοῦ πράγματος
γενέσθαι, οἵοίπερ ἂν ὑμῖν αὐτοῖς εἶητε 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 (LA)TEX 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` *New feature* v1.5

Inserting `xml:id`*s* 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`[101] and the like so as to insert them in the second argument—⟨*TEI pattern*⟩—of `\TeXtoTEIPat`. As an illustration, what follows creates a new LaTeX command `\witStart` which prints nothing in the PDF ouput but inserts `<witStart/>` elements in the `TEI xml` file. This commands accepts an optional argument to make clear which are the fragmentary witnesses involved:—

```
1 \NewDocumentCommand{\witStart}{O{}}{\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.[102]

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.

14.5 Inserting Code in the `TEI xml` Ouput 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` ouput.[103]

`\teidirect` *New feature* v1.3

`\teidirect[`⟨*xml attributes*⟩`]{`⟨*xml element*⟩`}{`⟨*code*⟩`}`

Two mandatory arguments are expected by `\teidirect`, namely the `TEI xml` element followed by the contents to be found in the ouput file between the opening and closing tags.

101. See above on page 8.
102. The LaTeX Team, *The Xparse package* [A generic document command parser] (Jan. 12, 2022), https://ctan.org/pkg/xparse.
103. See above sect. 2.5 on page 13.

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 *New feature* v1.5

\teidirectE[⟨*xml attributes*⟩]{⟨*xml element*⟩}

\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:

```
% Preamble:
\DeclareWitness{GalE1.M}{Gal.E1(M)}{\emph{Monacencis Gr.}
  231}[origDate=s. XVI]
\DeclareWitness{GalE1.Q}{Gal.E1(Q)}{\emph{Parisinus Gr.}
  2174}[origDate=s. XIV]
  \DeclareShorthand{GalE1.M.Q}{Gal.E1(MQ)}{GalE1.M,GalE1.Q}

% Document:
ἐν \app{
    \lem[wit=codd]{καύμασιν}
    \rdg[wit=GalE1.M.Q, postwit=\unskip(23.16)]{καύματι}
    \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 ouput file (l. 12).

PDF output:—

```
ἐν καύμασιν
```

1 καύμασιν codd.] καύματι Gal.E1(MQ)(23.16)

TEI xml output:—

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

14.6 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.[104] The technique applied below uses \EnvtoTEI conjointly with \SetTEIxmlExport{autopar=false} described above on page 76:—

```
% Preamble:
\newenvironment{speech}{\par}{\par}
\newcommand{\speaker}[1]{\textbf{#1}\par}
\EnvtoTEI{speech}{sp}

\SetTEIxmlExport{autopar=false}
```

104. See https://tei-c.org/release/doc/tei-p5-doc/en/html/TS.html.

```
7
8  % Document:
9  \begin{ekdosis}
10   \begin{speech}
11     \speaker{Σωκράτης}
12     \begin{ekdpar}
13       κατέβην χθὲς εἰς Πειραιᾶ μετὰ Γλαύκωνος τοῦ Ἀρίστωνος
14       προσευξόμενός τε τῇ θεῷ καὶ ἅμα τὴν ἑορτὴν βουλόμενος θεάσασθαι
15       τίνα τρόπον ποιήσουσιν ἅτε νῦν πρῶτον ἄγοντες. καλὴ μὲν οὖν μοι
16       καὶ ἡ τῶν ἐπιχωρίων πομπὴ ἔδοξεν εἶναι, οὐ μέντοι ἧττον ἐφαίνετο
17       πρέπειν ἣν οἱ Θρᾷκες 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` LaTeX 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:—

Σωκράτης

κατέβην χθὲς εἰς Πειραιᾶ μετὰ Γλαύκωνος τοῦ Ἀρίστωνος προσευξόμενός τε τῇ θεῷ καὶ ἅμα τὴν ἑορτὴν βουλόμενος θεάσασθαι τίνα τρόπον ποιήσουσιν ἅτε νῦν πρῶτον ἄγοντες. καλὴ μὲν οὖν μοι καὶ ἡ τῶν ἐπιχωρίων πομπὴ ἔδοξεν εἶναι, οὐ μέντοι ἧττον ἐφαίνετο πρέπειν ἣν οἱ Θρᾷκες ἔπεμπον.

`TEI xml` output:—

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

14.7 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(LA)TEX entry and its TEI equivalent are provided:[105]—

```
@Book{Drak,
  title = {Punicorum Libri Septemdecim},
  author = {Silius Italicus, Tiberius Catius},
  editor = {Drakenborch, Arnold},
  date = {1717},
  publisher = {Trajecti ad Rhenum},
  location = {Utrecht}
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<listBibl xmlns="http://www.tei-c.org/ns/1.0">
  <biblStruct type="book" xml:id="Drak">
    <monogr>
      <title level="m">Punicorum libri septemdecim</title>
      <author>
        <forename>Tiberius Catius</forename>
        <surname>Silius Italicus</surname>
      </author>
      <editor>
        <forename>Arnold</forename>
        <surname>Drakenborch</surname>
      </editor>
      <imprint>
        <pubPlace>Utrecht</pubPlace>
        <publisher>Trajecti ad Rhenum</publisher>
        <date>1717</date>
      </imprint>
    </monogr>
  </biblStruct>
</listBibl>
```

As can be seen, the same string `Drak` is used as a label in the Bib(LA)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,[106] like so:—

```
% Use 'bibl.xml' as a TEI xml bibliographical database:
\AddxmlBibResource{bibdata.xml}

% Declare A. Drakenborch as source:
\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):—

```
% Preamble:
\usepackage[style=oxnotes]{biblatex}
\addbibresource{bibdata.bib}

\usepackage[teiexport=tidy]{ekdosis}

```

105. To the author's knowledge, Zotero (https://www.zotero.org) provides excellent `TEI xml` output from Bib(LA)TEX input files.
106. See above on page 10.

```
% basename of the .xml bibliographical database:
\AddxmlBibResource{bibdata.xml}

% Witnesses:
\DeclareWitness{L}{L}{Laurentianus, plut, XXXVII, cod. 16}[
                      origDate=s. XV]
% Other witnesses [...]

% Sources:
\DeclareSource{Drak}{\emph{Drakenborch}}
% Alternatively, use BibLaTeX for the rendition:
% \DeclareSource{Drak}{\citename{Drak}{editor}}
% Other sources [...]

% Document:
\begin{ekdosis}
  \begin{ekdverse}
    Sed uos, quorum oculos atque ora humentia uidi,\\
    uertere cum consul terga et remeare iuberet,\\
    \app{
      \lem[source=Drak, type=emendation]{ne morem}
      \rdg[wit={L, F}]{me morem}
      \rdg[wit={O, V}]{memorem}
    } et pugnae signum exspectate petendae:\\
  \end{ekdverse}
\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):—

```
<text>
  <body>
    <lg>
      <l>Sed uos, quorum oculos atque ora humentia uidi,</l>
      <l>uertere cum consul terga et remeare iuberet,</l>
      <l>
      <app>
        <lem source="#Drak" type="emendation">ne morem</lem>
        <rdg wit="#L #F">me morem</rdg>
        <rdg wit="#O #V">memorem</rdg>
      </app>et pugnae signum exspectate petendae:</l>
    </lg>
  </body>
  <back>
    <listBibl>
      <biblStruct type="book" xml:id="Drak">
        <monogr>
          <title level="m">Punicorum libri septemdecim</title>
          <author>
            <forename>Tiberius Catius</forename>
```

```
            <surname>Silius Italicus</surname>
          </author>
          <editor>
            <forename>Arnold</forename>
            <surname>Drakenborch</surname>
          </editor>
          <imprint>
            <pubPlace>Utrecht</pubPlace>
            <publisher>Trajecti ad Rhenum</publisher>
            <date>1717</date>
          </imprint>
        </monogr>
      </biblStruct>
    </listBibl>
  </back>
</text>
```

14.8 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, BibTEX or BibLATEX 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(LA)TEX entries that are used in the document are found within `<biblStruct>` elements.[107] Finally, this `.xml` bibliographical database must have been connected to the `.tex` source file by means of `\AddxmlBibResource` described above in sect. 14.7 on page 83.

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>
        <forename>L. D.</forename>
        <surname>Reynolds</surname>
      </author>
```

107. See above n. 105 on page 84 for information on how to do this.

```
      <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[teiexport=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:—

On textual criticism, see L. D. Reynolds and N. G. Wilson, *Scribes and Scholars: A Guide to the Translation of Greek and Latin Literature* (3rd edn., Oxford: Clarendon Press, 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 xml` by ekdosis:—

(a) `\icite[`⟨*prenote*⟩`][`⟨*postnote*⟩`]{`⟨*key*⟩`}`[108]
(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”.

15 Future Work

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

(a) Very short-term (weeks):—
    i. Text structure: milestone elements.
    ii. 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.
(b) Short-term (months):—
    i. 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. Other

108. From the icite package. `\icite` can be used in place of almost any standard citation command. See Alessi, *The Icite package* (cf. n. 97).

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`.

ii. 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`.

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 LuaLaTeX 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), jdmdh: 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.

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

Bezos 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.

Knuth, Donald E., *The TEXBook* (32nd edn., Reading, Mass.: Addison–Wesley, 2013).

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.

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 3.70) [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.

Mittelbach, Frank, and Fairbairns, Robin, *The Footmisc package* (version 6.0e) [A range of footnote options] (May 26, 2022), https://ctan.org/pkg/footmisc.

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.

Tómács, Tibor, *The Fgruler package* (version 1.5) [Draw rulers on the foreground or in the text] (June 25, 2022), https://ctan.org/pkg/fgruler.

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

```
\documentclass[12pt]{article}
\usepackage{fontspec}
\usepackage[latin.classic,french,english]{babel}
\babelfont{rm}{Old Standard}
\babelfont{sf}{NewComputerModernSans10}
\babelfont{tt}{NewComputerModernMono10}

\usepackage{nextpage}
\usepackage{xltabular}

\usepackage[teiexport=tidy]{ekdosis}
\DeclareApparatus{default}[
        delim=\hskip0.75em,
        ehook=.]

\FormatDiv{2}{}{.}

\SetAlignment{
  tcols=3,
  lcols=1,
  texts=latin[xml:lang="la"];
        english[xml:lang="en"];
        french[xml:lang="fr"],
  apparatus=latin,
  segmentation=auto}

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

\DeclareWitness{A}{A}{\emph{Bongarsianus} 81}[
               msName=Bongarsianus,
               settlement=Amsterdam,
               idno=81,
               institution=University Library,
               origDate=s. IX--X]
\DeclareHand{A1}{A}{A\textsuperscript{1}}[\emph{Emendationes
               scribae ipsius}]
\DeclareWitness{M}{M}{\emph{Parisinus Lat.} 5056}[
               origDate={s. XII}]
\DeclareWitness{B}{B}{\emph{Parisinus Lat.} 5763}[
               origDate={s. IX--X}]
\DeclareWitness{R}{R}{\emph{Vaticanus Lat.} 3864}[
               origDate={s. X}]
\DeclareWitness{S}{S}{\emph{Laurentianus} R 33}[
               origDate={s. X}]
\DeclareWitness{L}{L}{\emph{Londinensis} Br. Mus. 10084}[
               origDate={s. XI}]
\DeclareWitness{N}{N}{\emph{Neapolitanus} IV, c. 11}[
               origDate={s. XII}]
\DeclareWitness{T}{T}{\emph{Parisinus Lat.} 5764}[
               origDate={s. XI}]
```

```
\DeclareWitness{f}{\emph{f}}{\emph{Vindobonensis} 95}[
               origDate={s. XII}]
\DeclareWitness{U}{U}{\emph{Vaticanus Lat.} 3324}[
               origDate={s. XI}]
\DeclareWitness{l}{\emph{l}}{\emph{Laurentianus} Riccard. 541}[
               origDate={s. XI--XII}]
\DeclareShorthand{a}{α}{A,M,B,R,S,L,N}
\DeclareShorthand{b}{β}{T,f,U,l}

\begin{document}

\begin{xltabular}[c]{0.75\linewidth}{lXl}
  \caption*{\textbf{Conspectus siglorum}\label{tab:conspectus-siglorum}}\\
  \multicolumn{3}{c}{\emph{Familia} \getsiglum{a}}\\
  \SigLine{A}\\
  & \getsiglum{A1} \emph{Emendationes scribae ipsius} & \\
  \SigLine{M}\\
  \SigLine{B}\\
  \SigLine{R}\\
  \SigLine{S}\\
  \SigLine{L}\\
  \SigLine{N}\\
  \multicolumn{3}{c}{\emph{Familia} \getsiglum{b}}\\
  \SigLine{T}\\
  \SigLine{f}\\
  \SigLine{U}\\
  \SigLine{l}\\
\end{xltabular}

\cleartoevenpage

\begin{alignment}
  \begin{latin}
    \ekddiv{head=XIII, depth=2, n=6.13, type=section}
    In omni Gallia eorum hominum qui \app{
      \lem[wit=a]{aliquo}
      \rdg[wit=b, alt=in al-]{in aliquo}}
    sunt numero atque honore genera sunt duo. Nam plebes paene
    seruorum habetur loco, quae \app{
      \lem[wit={A,M}, alt={nihil audet (aut et \getsiglum{A1})
        per se}]{nihil audet per se}
      \rdg[wit=A1,nordg]{nihil aut et per se}
      \rdg[wit={R,S,L,N}]{nihil habet per se}
      \rdg[wit=b]{per se nihil audet}}, \app{
      \lem[wit=a]{nullo}
      \rdg[wit=b]{nulli}} adhibetur \app{
      \lem{consilio}
      \rdg[wit={T, U}, alt=conc-]{concilio}}.
  \end{latin}
  \begin{english}
    \ekddiv{head=XIII, depth=2, n=6.13, type=section}
    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.
```

```
  \end{english}
  \begin{french}
    \ekddiv{head=XIII, depth=2, n=6.13, type=section}
    Partout en Gaule il y a deux classes d'hommes qui comptent et qui
    sont considérés. Quant aux gens du peuple, ils ne sont guère
    traités autrement que des esclaves, ne pouvant se permettre aucune
    initiative, n'étant consultés sur rien.
  \end{french}
\end{alignment}

\end{document}
```

17.2 TEI xml Output

```
<?xml version="1.0" encoding="utf-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>
          <!-- Title -->
        </title>
        <respStmt>
          <resp>
            <!-- Edited by -->
          </resp>
          <name>
            <!-- Name -->
          </name>
        </respStmt>
      </titleStmt>
      <publicationStmt>
        <distributor>
          <!-- Distributor name  -->
        </distributor>
      </publicationStmt>
      <sourceDesc>
        <listWit>
          <witness xml:id="A">
          <abbr type="siglum">A</abbr>
          <emph>Bongarsianus</emph>81
          <msDesc>
            <msIdentifier>
              <settlement>Amsterdam</settlement>
              <institution>University Library</institution>
              <idno>81</idno>
              <msName>Bongarsianus</msName>
            </msIdentifier>
            <physDesc>
              <handDesc hands="1">
                <handNote xml:id="A1">
                  <abbr type="siglum">A
                  <hi rend="sup">1</hi></abbr>
                  <p>
                    <emph>Emendationes scribae ipsius</emph>
```

```
                </p>
              </handNote>
            </handDesc>
          </physDesc>
          <history>
            <origin>
              <origDate>s. IX--X</origDate>
            </origin>
          </history>
        </msDesc></witness>
        <witness xml:id="M">
        <abbr type="siglum">M</abbr>
        <emph>Parisinus Lat.</emph>5056
        <msDesc>
          <msIdentifier />
          <history>
            <origin>
              <origDate>s. XII</origDate>
            </origin>
          </history>
        </msDesc></witness>
        <witness xml:id="B">
        <abbr type="siglum">B</abbr>
        <emph>Parisinus Lat.</emph>5763
        <msDesc>
          <msIdentifier />
          <history>
            <origin>
              <origDate>s. IX--X</origDate>
            </origin>
          </history>
        </msDesc></witness>
        <witness xml:id="R">
        <abbr type="siglum">R</abbr>
        <emph>Vaticanus Lat.</emph>3864
        <msDesc>
          <msIdentifier />
          <history>
            <origin>
              <origDate>s. X</origDate>
            </origin>
          </history>
        </msDesc></witness>
        <witness xml:id="S">
        <abbr type="siglum">S</abbr>
        <emph>Laurentianus</emph>R 33
        <msDesc>
          <msIdentifier />
          <history>
            <origin>
              <origDate>s. X</origDate>
            </origin>
          </history>
        </msDesc></witness>
        <witness xml:id="L">
```

```
          <abbr type="siglum">L</abbr>
          <emph>Londinensis</emph>Br. Mus. 10084
          <msDesc>
            <msIdentifier />
            <history>
              <origin>
                <origDate>s. XI</origDate>
              </origin>
            </history>
          </msDesc></witness>
          <witness xml:id="N">
          <abbr type="siglum">N</abbr>
          <emph>Neapolitanus</emph>IV, c. 11
          <msDesc>
            <msIdentifier />
            <history>
              <origin>
                <origDate>s. XII</origDate>
              </origin>
            </history>
          </msDesc></witness>
          <witness xml:id="T">
          <abbr type="siglum">T</abbr>
          <emph>Parisinus Lat.</emph>5764
          <msDesc>
            <msIdentifier />
            <history>
              <origin>
                <origDate>s. XI</origDate>
              </origin>
            </history>
          </msDesc></witness>
          <witness xml:id="f">
          <abbr type="siglum">
            <emph>f</emph>
          </abbr>
          <emph>Vindobonensis</emph>95
          <msDesc>
            <msIdentifier />
            <history>
              <origin>
                <origDate>s. XII</origDate>
              </origin>
            </history>
          </msDesc></witness>
          <witness xml:id="U">
          <abbr type="siglum">U</abbr>
          <emph>Vaticanus Lat.</emph>3324
          <msDesc>
            <msIdentifier />
            <history>
              <origin>
                <origDate>s. XI</origDate>
              </origin>
            </history>
```

```
        </msDesc></witness>
        <witness xml:id="l">
        <abbr type="siglum">
          <emph>l</emph>
        </abbr>
        <emph>Laurentianus</emph>Riccard. 541
        <msDesc>
          <msIdentifier />
          <history>
            <origin>
              <origDate>s. XI--XII</origDate>
            </origin>
          </history>
        </msDesc></witness>
      </listWit>
    </sourceDesc>
  </fileDesc>
  <encodingDesc>
    <variantEncoding method="parallel-segmentation"
    location="internal" />
  </encodingDesc>
</teiHeader>
<text>
  <body>
    <div xml:id="div-latin_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 #l">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 #l">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-english_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>
      <div xml:id="div-french_1" xml:lang="fr">
        <div type="section" n="6.13">
          <head>XIII</head>
          <p>Partout en Gaule il y a deux classes d'hommes qui
          comptent et qui sont considérés. Quant aux gens du
          peuple, ils ne sont guère traités autrement que des
          esclaves, ne pouvant se permettre aucune initiative,
          n'étant consultés sur rien.</p>
        </div>
      </div>
    </body>
  </text>
</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[teiexport=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.

```
\RequirePackage{iftex}
```

Of course, ekdosis requires LuaLaTeX. Issue an error if the document is processed with another engine.

```
\RequireLuaTeX
```

Set global options:—

```
\RequirePackage{expkv-opt}
\RequirePackage{expkv-def}
\newif\if@pkg@float
\newif\if@pkg@footins
\newif\if@pkg@keyfloat
\newif\if@pkg@fitapp
\newif\if@pkg@breakable
\newif\if@pkg@ekddivs
\newif\if@parnotesroman
\newif\if@pkg@parnotes
\newif\ifteei@export
\newif\if@pkg@poetry@verse
\ekvdefinekeys{ekdosis}{
  choice layout = {float = {\@pkg@floattrue},
    footins = {\@pkg@floatfalse\@pkg@footinstrue},
    keyfloat = {\@pkg@floatfalse\@pkg@keyfloattrue},
    fitapp = {\@pkg@floatfalse\@pkg@fitapptrue},
    breakable = {\@pkg@floatfalse\@pkg@fitapptrue\@pkg@breakabletrue}},
  initial layout = float,
  unknown-choice layout = \PackageError{ekdosis}{unknown
    layout=#1}{`layout' must be either `float', `footins', `keyfloat',
    `fitapp' or `breakable'.},
  choice divs = {ekdosis = {\@pkg@ekddivstrue},
    latex = {\@pkg@ekddivsfalse
      \AtBeginDocument{\luadirect{ekdosis.setekddivsfalse()}}}},
  initial divs = ekdosis,
  unknown-choice divs = \PackageError{ekdosis}{unknown divs=#1}{`divs'
    must be either `ekdosis' or `latex'.},
  choice poetry = {verse = {\@pkg@poetry@versetrue}},
  unknown-choice poetry = \PackageError{ekdosis}{unknown
    poetry=#1}{`poetry' must be `verse' for now.},
  choice parnotes = {false = {},
    true = {\@pkg@parnotestrue},
    roman = {\@pkg@parnotestrue\@parnotesromantrue}},
  default parnotes = true,
  unknown-choice parnotes = \PackageError{ekdosis}{unknown
    parnotes=#1}{`parnotes' must be either `true', or `false' or
    `roman'.},
```

```
  choice teiexport = {false = {},
    true = {\tei@exporttrue
      \AtBeginDocument{\luadirect{ekdosis.openteistream()}}%
      \AtEndDocument{\luadirect{ekdosis.closeteistream()}}},
    tidy = {\tei@exporttrue
      \AtBeginDocument{\luadirect{ekdosis.openteistream()}}%
      \AtEndDocument{\luadirect{ekdosis.closeteistream("tidy")}}}},
  default teiexport = true,
  unknown-choice teiexport = \PackageError{ekdosis}{unknown
    teiexport=#1}{`teiexport' must be either `true', `false' or
    `tidy'.}
}
\ekvoProcessLocalOptions{ekdosis}
\newif\ifekd@memoir@loaded
\@ifclassloaded{memoir}{%
  \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:—

```
% \RequirePackage{iftex} % already loaded above
% \RequirePackage{expkv-opt} % already loaded above
% \RequirePackage{expkv-def} % already loaded above
\RequirePackage{luacode}
\RequirePackage{paracol}
\RequirePackage{etoolbox}
\RequirePackage{lineno}
\if@pkg@float
  \RequirePackage{trivfloat}
  \trivfloat{ekdapparatus}
\fi
\if@pkg@keyfloat
  \RequirePackage{keyfloat}
  \def\ekd@keyparopts#1{%
    \def\ekd@insert@keyparapp{%
      \keyparbox[!b]{#1}{\ekd@insert@apparatus}}}
  \ekd@keyparopts{}
\fi
\if@pkg@fitapp
  \RequirePackage{tcolorbox}
  \tcbuselibrary{fitting,skins}
\fi
\if@pkg@breakable
  \RequirePackage{tcolorbox}
  \tcbuselibrary{fitting,skins,breakable}
\fi
\RequirePackage{refcount}
\RequirePackage{zref-user}
\RequirePackage{zref-abspage}
\RequirePackage{ltxcmds}
\RequirePackage{pdftexcmds}
\RequirePackage{ifoddpage}
\if@pkg@poetry@verse
  \RequirePackage{verse}
\fi
\if@pkg@parnotes
```

```
  \RequirePackage{parnotes}
\fi
```

Lua Here begins the real work: load `ekdosis.lua`:—

```
\luadirect{dofile(kpse.find_file("ekdosis.lua"))}
\AtEndDocument{
  \luadirect{ekdosis.closestream()}
}
```

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.

```
\def\eKd{%
  ε\kern -.4em\raise 1.15ex\hbox{κ}\kern -.105emδ%
  \ifdefined\xspace\xspace\fi
}
```

Setup

`\ekdsetup` `\ekdsetup` is used to specify options that affect the general behavior of ekdosis. It is a preamble-only command.

```
\ekvdefinekeys{ekd@setup}{
  bool showpagebreaks = \ifekd@showpb,
  store spbmk = \ekd@spbmk,
  initial spbmk = spb,
  store hpbmk = \ekd@hpbmk,
  initial hpbmk = hpb
}
\NewDocumentCommand{\ekdsetup}{m}{\ekvset{ekd@setup}{#1}}
\@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.

```
\ekvdefinekeys{ekd@hooks}{
  store appfontsize = \ekd@appfontsize,
  store refnumstyle = \ekd@refnumstyle,
  store postrefnum = \ekd@postrefnum,
  code familysep = \luadirect{ekdosis.setfamilysep(\luastringN{#1})},
  store lemmastyle = \ekd@lemmastyle,
  store readingstyle = \ekd@readingstyle,
  code initialrule = \def\ekd@initial@rule{#1\NLS},
  default initialrule = \rule{0.4\columnwidth}{0.4pt},
  noval noinitialrule = \undef\ekd@initial@rule,
  code maxentries = \luadirect{ekdosis.setglimit(\luastringN{#1})},
  nmeta nomaxentries = {maxentries=none},
  code keyparopts = \if@pkg@keyfloat\ekd@keyparopts{#1}\fi,
  dimen appheight = \ekd@app@height,
  initial appheight = .5\textheight,
  choice fitalgorithm = {fontsize = \def\ekd@fit@algorithm{fontsize},
    hybrid = \def\ekd@fit@algorithm{hybrid},
    areasize = \def\ekd@fit@algorithm{areasize},
```

```
    squeeze = \def\ekd@fit@algorithm{squeeze}},
  initial fitalgorithm = fontsize,
  unknown-choice fitalgorithm = \PackageError{ekdosis}{unknown
    fitalgorithm=#1}{`fitalgorithm' must be either `fontsize',
    `hybrid', `areasize' or `squeeze'.},
  initial appfontsize = \footnotesize,
  initial refnumstyle = \bfseries,
  initial postrefnum = ~,
  initial lemmastyle = {},
  initial readingstyle = {}
}
\NewDocumentCommand{\SetHooks}{m}{\ekvset{ekd@hooks}{#1}}
```

Build and process the list of witnesses and hands:—

```
\ekvdefinekeys{ekd@witness}{
  store settlement = \settlement@value,
  store institution = \institution@value,
  store repository = \repository@value,
  store collection = \collection@value,
  store idno = \idno@value,
  store msName = \msName@value,
  store origDate = \origDate@value,
  store locus = \locus@value,
  store additional = \additional@value
}
```

\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.

```
\NewDocumentCommand{\DeclareWitness}{m m m O{}}{%
  \bgroup
  \ekvset{ekd@witness}{#4}
    \luadirect{ekdosis.newwitness(
      \luastringN{#1},
      \luastringN{#2},
      \luastringN{#3},
      \luastringO{\settlement@value},
      \luastringO{\institution@value},
      \luastringO{\repository@value},
      \luastringO{\collection@value},
      \luastringO{\idno@value},
      \luastringO{\msName@value},
      \luastringO{\origDate@value},
      \luastringO{\locus@value},
      \luastringO{\additional@value})}
    \egroup
  }
\@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.

```
\NewDocumentCommand{\DeclareHand}{m m m +O{}}{
```

```
  \luadirect{ekdosis.newhand(\luastringN{#1},
    \luastringN{#2},
    \luastringN{#3},
    \luastringN{#4})}
}
\@onlypreamble\DeclareHand
```

Build and process the list of scholars:—

```
\ekvdefinekeys{ekd@scholar}{
  store rawname = \rawname@value,
  store forename = \forename@value,
  store surname = \surname@value,
  store addname = \addname@value,
  store note = \note@value
}
```

\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.

```
\NewDocumentCommand{\DeclareScholar}{m m O{}}{%
  \bgroup
  \ekvset{ekd@scholar}{#3}
    \luadirect{ekdosis.newscholar(
      \luastringN{#1},
      \luastringN{#2},
      \luastringO{\rawname@value},
      \luastringO{\forename@value},
      \luastringO{\surname@value},
      \luastringO{\addname@value},
      \luastringO{\note@value})}
    \egroup
  }
\@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(LA)TEX label) to be used in the apparatus criticus which can be extracted from a bibliographic database.

```
\NewDocumentCommand{\DeclareSource}{m m}{
  \luadirect{ekdosis.newsource(\luastringN{#1},
    \luastringN{#2})}
}
\@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.

```
\NewDocumentCommand{\DeclareShorthand}{m m m}{
  \luadirect{ekdosis.newshorthand(\luastringN{#1},
    \luastringN{#2},
```

```
    \luastringN{#3})}
}
\@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.

```
\NewDocumentCommand{\getsiglum}{m}{%
  \luadirect{tex.sprint(ekdosis.getsiglum(\luastringN{#1}))}%
}
```

\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.

```
\NewDocumentCommand{\SigLine}{m}{%
  \luadirect{tex.sprint(ekdosis.basic_cs(\luastringN{#1}))}
}
```

TEX to `TEI xml` Here follow the `key-value` options to be used by `\SetTEIxmlExport` below:—

```
\ekvdefinekeys{tei@settings}{
  choice autopar = {true = \luadirect{ekdosis.setteiautopar("yes")},
    false = {\luadirect{ekdosis.setteiautopar("no")}}},
  initial autopar = true,
  unknown-choice autopar = \PackageError{ekdosis}{unknown
    autopar=#1}{`autopar' must be either `true' or `false'.}
}
```

\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.

```
\NewDocumentCommand{\SetTEIxmlExport}{m}{
  \unless\ifekd@state\ekvset{tei@settings}{#1}\fi
}
```

The following three commands can be used to instruct ekdosis how to convert unknown or unusual (LA)TEX 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:—

```
\NewDocumentCommand{\TeXtoTEI}{m m O{}}{%
  \luadirect{ekdosis.newcmdtotag(\luastringN{#1},
    \luastringN{#2},
    \luastringN{#3})}
}
```

\teidirect `\teidirect[`⟨*xml attributes*⟩`]{`⟨*xml element*⟩`}{`⟨*code*⟩`}` does nothing in LATEX. It is only
\teidirectE 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.

```
\NewDocumentCommand{\teidirect}{O{} m m}{\ignorespaces}
\NewDocumentCommand{\teidirectE}{O{} 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).

```
\NewDocumentCommand{\getTEIxmlid}{m}{%
  \luadirect{tex.sprint(ekdosis.getsiglum(\luastringN{#1}, "TEI"))}%
}
```

`\EnvtoTEI` `\EnvtoTEI`⟨`*`⟩`{`⟨*env name*⟩`}{`⟨*TEI element*⟩`}[`⟨*TEI attribute(s)*⟩`]` 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.

```
\NewDocumentCommand{\EnvtoTEI}{s m m O{}}{%
  \IfBooleanTF{#1}{%
  \luadirect{ekdosis.newenvtotag(\luastringN{#2},
    \luastringN{#3},
    \luastringN{#4},
    "yes")}
  }{%
  \luadirect{ekdosis.newenvtotag(\luastringN{#2},
    \luastringN{#3},
    \luastringN{#4})}
  }
}
```

`\EnvtoTEI*`

`\TeXtoTEIPat` Finally, the more flexible—and more delicate to handle—`\TeXtoTEIPat{`⟨*TEX pattern*⟩`}{`⟨*TEI pattern*⟩`}` uses pattern matching to instruct ekdosis how to convert (LaTeX commands into `TEI` equivalents.

```
\NewDocumentCommand{\TeXtoTEIPat}{m m}{%
  \luadirect{ekdosis.newpatttotag(\luastringN{#1}, \luastringN{#2})}
}
```

`\SetTEIFilename` `\SetTEIFilename{`⟨*basename*⟩`}` 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`:—

```
\NewDocumentCommand{\SetTEIFileName}{m}{
  \luadirect{ekdosis.setteifilename(\luastringN{#1})}
}
\@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{`⟨*basename or name.xml*⟩`}`, 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.

```
\NewDocumentCommand{\AddxmlBibResource}{m}{
  \luadirect{ekdosis.addxmlbibresource(\luastringN{#1})}
}
\@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 `\languagename` can be inserted when and where needed in the apparatus criticus.

```
\newif\ifekd@lang@pkg
\NewDocumentCommand{\ekd@test@lang}{}{%
  \ltx@ifpackageloaded{babel}{\ekd@lang@pkgtrue}{}%
  \ltx@ifpackageloaded{polyglossia}{\ekd@lang@pkgtrue}{}%
}
```

Multiple-layer apparatuses ekdosis must know if an entry is to be processed in a single- or multiple-layer context:—

```
\newif\ifekd@mapps
\newif\ifekdl@nowit
```

Now the key-value options can be defined:—

```
\ekvdefinekeys{ekd@newapp}{
  choice direction = {LR = \def\direction@val{LR},
                      RL = \def\direction@val{RL}},
  unknown-choice direction = \PackageError{ekdosis}{unknown
    direction=#1}{`direction' must be either `LR' or `RL'.},
  store rule = \rule@val,
  nmeta norule = {rule=none},
  code delim = \def\delim@val{\unexpanded{#1}},
  store sep = \sep@val,
  store subsep = \subsep@val,
  store bhook = \bhook@val,
  store ehook = \ehook@val,
  store maxentries = \limit@val,
  store lang = \lang@val,
  store notelang = \notelang@val,
  initial direction = LR,
  initial delim = {},
  initial ehook = {\csname ekd@end@apparatus\endcsname},
  choice negative = {true=\def\negative@val{yes},
    false=\def\negative@val{no}},
  initial negative = false,
  default negative = true,
  unknown-choice negative = \PackageError{ekdosis}{unknown
    negative=#1}{`negative' must be either `true' or `false'.},
  choice TEInegative = {true=\def\TEInegative@val{yes},
    false=\def\TEInegative@val{no}},
  initial TEInegative = false,
  default TEInegative = true,
  unknown-choice TEInegative = \PackageError{ekdosis}{unknown
    TEInegative=#1}{`TEInegative' must be either `true' or `false'.}
}
```

\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 eleven key-value options can be used to lay out the layer: `direction=LR|RL`, `rule` `norule`, `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), `lang`,

`notelang` (the language to be applied) `negative`, `TEInegative` (the form of the apparatus criticus, either positive or negative):—

```
\NewDocumentCommand{\DeclareApparatus}{m O{}}{
  \newbool{subsq@unit@#1}
  \booltrue{subsq@unit@#1}
  \newbool{ekdl@nowit@#1}
  \unless\ifekd@mapps\global\ekd@mappstrue\fi
  \bgroup
  \ekvset{ekd@newapp}{#2}
  \luadirect{ekdosis.newapparatus(
    \luastringN{#1},
    \luastring{\direction@val},
    \luastringO{\rule@val},
    \luastringO{\delim@val},
    \luastringO{\sep@val},
    \luastringO{\subsep@val},
    \luastringO{\bhook@val},
    \luastringO{\ehook@val},
    \luastringO{\limit@val},
    \luastringO{\lang@val},
    \luastringO{\notelang@val},
    \luastringO{\negative@val},
    \luastringO{\TEInegative@val}
    )}
  \egroup
}
\@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.

```
\NewDocumentCommand{\addentries}{O{\ekdan@type} m}{%
  \luadirect{ekdosis.addto_bagunits(\luastringO{#1}, \luastringN{#2})}%
  \ignorespaces
}
```

`\ekdpb` `\ekdpb[`⟨*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`.

```
\newcounter{ekd@pb}
\globalcounter{ekd@pb}
\NewDocumentCommand{\ekdpb}{s o m}{%
  \IfBooleanTF{#1}
  {\ifekd@showpb\marginpar{\ekd@hpbmk}\fi
    \pagebreak
    \@ifnextchar\bgroup{}{#3}%
  }
  {%
    \def\@tmpoarg{#2}%
    \def\@tmpmarg{#3}%
    \stepcounter{ekd@pb}%
    \linelabel{ekdpb:\theekd@pb}%
    \def\tmp@ln{%
```

```
      \getrefnumber{ekdpb:\theekd@pb}}%
    \def\tmp@pg{%
      \getpagerefnumber{ekdpb:\theekd@pb}}%
    \IfNoValueTF{#2}
    {\ifnum
      \pdf@strcmp{\@tmpmarg}{\tmp@ln} = 0
      \ifekd@showpb\marginpar{\ekd@spbmk}\fi
      \pagebreak
      \else
      \ifekd@showpb\marginpar{[\ekd@spbmk]}\fi
      \fi}
    {\ifnum
      \pdf@strcmp{\@tmpoarg}{\tmp@pg} = 0
      \ifnum
      \pdf@strcmp{\@tmpmarg}{\tmp@ln} = 0
      \ifekd@showpb\marginpar{\ekd@spbmk}\fi
      \pagebreak
      \else
      \ifekd@showpb\marginpar{[\ekd@spbmk]}\fi
      \fi
      \fi
    }%
  }\ignorespaces
}
```

Apparatus-related settings and functions. Some booleans to check if an apparatus should be inserted and what is the current environment.

```
\newbool{do@app}
\newif\ifekd@state
\newif\ifekd@isinapp
\newif\ifekd@isinlem
\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.

```
\providebool{al@rlmode}
\@ifpackageloaded{arabluatex}{}{%
  \def\setRL{\booltrue{al@rlmode}\pardir TRT\textdir TRT}
  \def\setLR{\boolfalse{al@rlmode}\pardir TLT \textdir TLT}
}
\protected\def\LRnum#1{\bgroup\textdir TLT#1\egroup}
```

Set a counter referring to line numbers and make it global.

```
\newcounter{ekd@lab}
\globalcounter{ekd@lab}
```

This command inserts words in the apparatus criticus without checking if both `ekd@isinapp` and `ekd@state` are set to `true`.

```
\NewDocumentCommand{\unconditional@appin}{o m}{%
  \IfNoValueTF{#1}
  {\luadirect{ekdosis.appin(\luastring0{#2})}}
  {\luadirect{ekdosis.appin(\luastring0{#2}, \luastring0{#1})}}%
}
```

`\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.

```
\def\blfootnote{\gdef\@thefnmark{\relax}\@footnotetext}
 % \def\blfootnote{\gdef\@thefnmark{}\@blfootnotetext}
\long\def\@blfootnotetext#1{\insert\footins{%
    \reset@font\footnotesize
    \interlinepenalty\interfootnotelinepenalty
    \splittopskip\footnotesep
    \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
    \hsize\columnwidth \@parboxrestore
    \protected@edef\@currentlabel{%
       \csname p@footnote\endcsname\@thefnmark
    }%
    \color@begingroup
      \@makeblfntext{%
        \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
    \color@endgroup}}%
\newcommand\@makeblfntext[1]{%
    \parindent 1em%
    \noindent
    \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:—

```
\newif\ifrtl@app
\def\ekdsep{] }
\def\ekdsubsep{}
\ekvdefinekeys{default@app}{
  choice direction = {LR = \rtl@appfalse,
    RL = \rtl@apptrue},
  unknown-choice direction = \PackageError{ekdosis}{unknown
    direction=#1}{`direction' must be either `LR' or `RL'.},
  code sep = \def\ekdsep{#1},
  code subsep = \def\ekdsubsep{#1},
  store bhook = \ekd@begin@apparatus,
  initial bhook = {},
  store ehook = \ekd@end@apparatus,
  initial ehook = {},
  store delim = \ekd@unit@delim,
  initial delim = {},
  store rule = \ekd@default@rule,
  initial rule = \rule{0.4\columnwidth}{0.4pt},
  noval norule = \def\ekd@default@rule{\mbox{}},
  store lang = \ekd@singleapp@lang,
  initial lang = \ltx@ifpackageloaded{babel}{\languagename}{%
    \ltx@ifpackageloaded{polyglossia}{\languagename}{}},
  store notelang = \ekd@singleapp@note@lang,
  initial notelang = \ltx@ifpackageloaded{babel}{\languagename}{%
    \ltx@ifpackageloaded{polyglossia}{\languagename}{}},
  choice negative = {true=\ekdl@nowittrue,
    false=\ekdl@nowitfalse},
  initial negative = false,
  default negative = true,
  unknown-choice negative = \PackageError{ekdosis}{unknown
    negative=#1}{`negative' must be either `true' or `false'.},
  choice TEInegative = {
    true=\luadirect{ekdosis.set_negpos_apparatus("neg")},
```

```
    false=\luadirect{ekdosis.set_negpos_apparatus("pos")}},
  initial TEInegative = false,
  default TEInegative = true,
  unknown-choice TEInegative = \PackageError{ekdosis}{unknown
    TEInegative=#1}{`TEInegative' must be either `true' or `false'.}
}
```

`\SetApparatus` All settings can also be defined as `key-value` options within the argument of `\SetApparatus`:—

```
\NewDocumentCommand{\SetApparatus}{m}{
  \ekvset{default@app}{#1}
}
```

`\SetLTRapp` `\SetRTLapp` `\SetLTRapp` and `\SetRTLapp` are two argument-less commands to set the direction of single-layer apparatus criticus, either LTR or RTL:—

```
\NewDocumentCommand{\SetRTLapp}{}{\rtl@apptrue}
\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 (`]␣`):—

```
\NewDocumentCommand{\SetSeparator}{m}{\def\ekdsep{#1}}
```

`\SetSubseparator` `\SetSubseparator{`⟨*sub-separator*⟩`}` allows to change the "subseparator" between variant readings. By default, no subseparator is set:—

```
\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:—

```
\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:—

```
\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):—

```
\NewDocumentCommand{\SetUnitDelimiter}{m}{\def\ekd@unit@delim{#1}}
```

`\SetApparatusLanguage` `\SetApparatusLang{`⟨*languagename*⟩`}` 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.

```
\NewDocumentCommand{\SetApparatusLanguage}{m}{%
  \def\ekd@singleapp@lang{#1}}
```

`\SetApparatusNoteLanguage` `\SetApparatusNoteLang{`⟨*languagename*⟩`}` 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.

```
\NewDocumentCommand{\SetApparatusNoteLanguage}{m}{%
  \def\ekd@singleapp@note@lang{#1}}
```

\footnoteruletrue
\footnoterulefalse

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 LaTeX "footnoterule" printed on every page of the edition text. `\footnoterulefalse` removes it while `\footnoteruletrue` leaves it untouched. The latter is set by default.

\SetNegativeApparatus
\SetPositiveApparatus

`\SetNegativeApparatus` and `\SetPositiveApparatus` are two argument-less commands. The former has the witnesses attached to the lemma text removed from the apparatus criticus in print, while the latter prints the apparatus criricus in positive form, which is the default.

\SetTEINegativeApparatus
\SetTEIPositiveApparatus

`\SetTEINegativeApparatus` and `\SetTEIPositiveApparatus` do the same as described above for the apparatus criticus converted in `TEI xml`.

```
\NewDocumentCommand{\SetNegativeApparatus}{}{\ekdl@nowittrue}
\NewDocumentCommand{\SetPositiveApparatus}{}{\ekdl@nowitfalse}
\NewDocumentCommand{\SetTEINegativeApparatus}{}{%
  \luadirect{ekdosis.set_negpos_apparatus("neg")}%
}
\NewDocumentCommand{\SetTEIPositiveApparatus}{}{%
  \luadirect{ekdosis.set_negpos_apparatus("pos")}%
}
```

```
\newif\iffootnoterule
\footnoteruletrue
\let\dflt@footnoterule\footnoterule
\let\dflt@pcol@footnoterule\pcol@footnoterule
\renewcommand\footnoterule{%
  \iffootnoterule
  \dflt@footnoterule%
  \fi
}
\renewcommand\pcol@footnoterule{%
  \iffootnoterule
  \dflt@pcol@footnoterule%
  \fi
}
```

\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.

```
\NewDocumentCommand{\SetDefaultRule}{m}{%
  \def\@tempa{#1}
  \ifx\@tempa\empty\def\ekd@default@rule{\mbox{}}%
  \else%
  \def\ekd@default@rule{#1}%
  \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.

```
\newcommand*{\NLS}{%
  \nobreak\@normalcr\relax
  % \par
  % \nobreak
```

```
  % \vspace{-\parskip}%
  % \leavevmode
  % \noindent
  % \ignorespaces
}
```

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.

```
\newif\ifsubsq@unit
\subsq@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.

```
\newif\ifekd@inside@app
\newif\ifekd@keepinapp
```

`\ekd@app@localheight` is used to set the maximum height of the apparatus block locally:—

```
\newlength{\ekd@app@localheight}
```

`\localappheight` `\localappheight{`⟨*dimen*⟩`}` can be used to change locally the length of `\ekd@app@height` set by the `appheight` option of `\SetHooks`, namely the height up to which the apparatus block is allowed to grow. ⟨*dimen*⟩ must be a number followed by a unit length. This command operates only on the apparatus block that follows it.

```
\def\localappheight#1{%
  \if@pkg@fitapp
    \luadirect{ekdosis.changeappheight()}%
    \setlength{\ekd@app@localheight}{#1}%
  \fi
  \ignorespaces
}
```

`\addtoappheight` As the name suggests, in contrast to `\localappheight`, `\addtoappheight{`⟨*dimen*⟩`}` is used to increase or decrease locally the length of `\ekd@app@height`. ⟨*dimen*⟩ must be a number followed by a unit length. This command operates only on the apparatus block that follows it.

```
\def\addtoappheight#1{%
  \if@pkg@fitapp
    \luadirect{ekdosis.changeappheight()}%
    \setlength{\ekd@app@localheight}{\ekd@app@height}%
    \addtolength{\ekd@app@localheight}{#1}%
  \fi
  \ignorespaces
}
```

Then `\ekd@fitapp` is defined for `layout=fitapp`:—

```
\if@pkg@fitapp
  \newtcboxfit{\ekd@fitapp}{%
    blankest,
    \if@pkg@breakable breakable\fi,
    fit basedim = \f@size pt,
    fit fontsize macros,
    fit height from=0pt to \ekd@app@height,
    fit algorithm = \ekd@fit@algorithm,
```

```
    float=!b}
\fi
```

Then `\ekd@breakable` for `layout=breakable`:—

```
% \if@pkg@breakable
%   \newtcboxfit{\ekd@breakable}{%
%     blankest,
%     breakable,
%     float=!b}
% \fi
```

Finally two commands are used to actually insert the apparatus depending on the value set in the `layout` global option.

```
\long\def\ekd@insert@apparatus{%
  \unless\ifekd@mapps
  \ifrtl@app\pardir TRT\leavevmode\textdir TRT\else
    \pardir TLT\leavevmode\textdir TLT\fi
  \fi
  \if@pkg@parnotes
   \if@parnotesroman
    \renewcommand*{\theparnotemark}{\roman{parnotemark}}\fi
   \parnoteclear\fi
  \ekd@inside@apptrue
  \ekd@appfontsize
  \ifekd@mapps
    \ifdefined\ekd@initial@rule
      \ekd@initial@rule
    \fi
  \fi
  \apparatus\unless\ifekd@mapps\ekd@end@apparatus\fi
  \ekd@inside@appfalse
  \if@pkg@parnotes\parnotes\parnotereset\fi
}
```

Depending on what is instructed, either of the following two commands is inserted by the Lua function `ekdosis.setheightandprintapparatus()` that is used in `\add@apparatus` below:—

```
\def\ekd@insert@fitapparatus@tmpheight{%
  \let\ekd@app@savedheight\ekd@app@height
  \let\ekd@app@height\ekd@app@localheight
  \ekd@fitapp{\ekd@insert@apparatus}%
  \let\ekd@app@height\ekd@app@savedheight}
\def\ekd@insert@fitapparatus{%
  \ekd@fitapp{\ekd@insert@apparatus}%
}
\def\add@@apparatus{%
  \if@pkg@parnotes\parnotes\else\fi
  \if@pkg@footins
  \bgroup
    \unless\ifekd@mapps
      \ifrtl@app\pardir TRT\leavevmode\textdir TRT\else
        \pardir TLT\leavevmode\textdir TLT\fi
    \fi
  \blfootnote{%
    \if@pkg@parnotes
    \if@parnotesroman
    \renewcommand*{\theparnotemark}{\roman{parnotemark}}\else\fi
```

```
    \parnoteclear\else\fi
    \ekd@inside@apptrue
    \ekd@appfontsize
    \ifekd@mapps
      \ifdefined\ekd@initial@rule
        \ekd@initial@rule
      \fi
    \fi
    \apparatus\unless\ifekd@mapps\ekd@end@apparatus\fi
    \ekd@inside@appfalse
    \if@pkg@parnotes\parnotes\parnotereset\else\fi
  }%
  \egroup
  \fi
  \if@pkg@float
  \begin{ekdapparatus}[!b]%
    \ekd@insert@apparatus
  \end{ekdapparatus}%
  \fi
  \if@pkg@keyfloat
    \ekd@insert@keyparapp
  \fi
  \if@pkg@fitapp
    \luadirect{tex.sprint(ekdosis.setheightandprintapparatus())}%
  \fi
  % \if@pkg@breakable
  %   \ekd@breakable{\ekd@insert@apparatus}%
  % \fi
}
```

Before inserting any new entry, `\add@apparatus` calls `\test@apparatus` to decide whether a new apparatus block must be created on a given page.

```
\def\add@apparatus{%
  \test@apparatus%
  \ifbool{do@app}{\subsq@unitfalse\add@@apparatus}{}%
}
```

`\append@app` inserts a bare (sub)entry in the apparatus...

```
\NewDocumentCommand{\append@app}{o +m}{%
  \ifekd@isinapp%
    \ifekd@state%
    \IfNoValueTF{#1}%
      {\luadirect{ekdosis.appin(\luastringO{#2})}}%
      {\luadirect{ekdosis.appin(\luastringO{#2}, \luastringO{#1})}}%
    \fi%
  \fi}
```

while `\append@ln@app` inserts a (sub)entry possibly preceded by a line number.

```
\NewDocumentCommand{\append@ln@app}{o o +m}{%
  \IfNoValueTF{#2}
  {\IfNoValueTF{#1}
    {\luadirect{tex.sprint(ekdosis.mdvappend(\luastringO{#3}))}}
    {\luadirect{tex.sprint(ekdosis.mdvappend(\luastringO{#3},
        \luastringO{#1}))}}%
  }
  {\IfNoValueTF{#1}
    {\luadirect{tex.sprint(ekdosis.mdvappend(\luastringO{#3}, nil,
```

```
      \luastringO{#2}))}}
  {\luadirect{tex.sprint(ekdosis.mdvappend(\luastringO{#3},
      \luastringO{#1},
      \luastringO{#2}))}}%
  }%
}
```

Lineation Settings

\outerlinenumbers
\innerlinenumbers

ekdosis does not use the "pagewise" numbering mode that is provided by lineno. Therefore, \outerlinenumbers and \innerlinenumbers are defined in addition to \rightlinenumbers and \leftlinenumbers.

```
\def\outerlinenumbers{%
  \def\makeLineNumberRunning{%
    \checkoddpage
    \ifoddpage
    \linenumberfont\hskip\linenumbersep\hskip\textwidth
    \hbox to\linenumberwidth{\hss\LineNumber}\hss
    \else
    \hss\linenumberfont\LineNumber\hskip\linenumbersep
    \fi
  }%
}
\def\innerlinenumbers{%
  \def\makeLineNumberRunning{%
    \checkoddpage
    \ifoddpage
    \hss\linenumberfont\LineNumber\hskip\linenumbersep
    \else
    \linenumberfont\hskip\linenumbersep\hskip\textwidth
    \hbox to\linenumberwidth{\hss\LineNumber}\hss
    \fi
  }%
}
```

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.

```
\newif\ifekd@pagelineation
\newif\ifekd@hidelinenumbers
\newif\ifekd@pagevlineation
```

Two counters (ekd@lnperpage and ekd@locallnperpage) are defined here and will be used below to allow ekdosis to have control over the maximum number of lines to be printed per page.

```
\newcounter{ekd@lnperpage}
\newcounter{ekd@locallnperpage}
\NewDocumentCommand{\ekdatbegshihook}{}{%
  \ifekd@pagelineation\resetlinenumber\fi
  \setcounter{ekd@lnperpage}{0}%
}
\AddToHook{shipout/before}{\ekdatbegshihook}
\newif\ifekd@elidednumbers
\ekvdefinekeys{ekd@lineation}{
  choice lineation = {page = \ekd@pagelineationtrue
                              \ekd@hidelinenumbersfalse,
```

```
    document = \ekd@pagelineationfalse
               \ekd@hidelinenumbersfalse,
    none = \ekd@pagelineationtrue
           \ekd@hidelinenumberstrue},
  unknown-choice lineation = \PackageError{ekdosis}{unknown
    lineation=#1}{`lineation' must be either `page' or `document'.},
  choice vlineation = {page = \ekd@pagevlineationtrue,
    document = \ekd@pagevlineationfalse},
  unknown-choice vlineation = \PackageError{ekdosis}{unknown
    vlineation=#1}{`vlineation' must be either `page' or `document'.},
  code modulonum = \chardef\c@linenumbermodulo#1\relax,
  noval modulo = \modulolinenumbers,
  code vmodulo = \ifekd@memoir@loaded\linenumberfrequency{#1}
                 \else\if@pkg@poetry@verse\poemlines{#1}\fi\fi,
  initial vmodulo = 1,
  default vmodulo = 5,
  bool vnumbrokenlines = \ifnum@brokenline,
  bool continuousvnum = \if@continuous@vnum,
  choice numbers = {elided = \ekd@elidednumberstrue,
    full = \ekd@elidednumbersfalse},
  unknown-choice numbers = \PackageError{ekdosis}{unknown
    numbers=#1}{`numbers' must be either `elided' or `full'.},
  initial numbers = elided,
  choice margin = {right = \rightlinenumbers,
    left = \leftlinenumbers,
    inner = \innerlinenumbers,
    outer = \outerlinenumbers},
  unknown-choice margin = \PackageError{ekdosis}{unknown
    margin=#1}{`margin' must be either `left', `right', \MessageBreak
    `inner' or `outer'},
  choice vmargin = {
    right = \if@pkg@poetry@verse\verselinenumbersright\fi,
    left = \if@pkg@poetry@verse\verselinenumbersleft\fi},
  unknown-choice vmargin = \PackageError{ekdosis}{unknown
    vmargin=#1}{`margin' must be either `left' ot `right'},
  code maxlines = \def\maxlines@value{#1},
  code nomaxlines = \undef\maxlines@value
}
```

\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.

```
\NewDocumentCommand{\SetLineation}{m}{
  \ekvset{ekd@lineation}{#1}
}
```

\vmodulolinenumbers

```
\NewDocumentCommand{\vmodulolinenumbers}{O{5}}{%
  \ifekd@memoir@loaded
    \linenumberfrequency{#1}%
  \else
    \if@pkg@poetry@verse
      \poemlines{#1}%
    \fi
  \fi
```

```
  \ignorespaces
}
```

Use `\normalfont` for line numbers:—

```
\renewcommand\linenumberfont{\normalfont\footnotesize}
```

Limiting the Number of Lines per Page The following commands are provided to set and control the maximum number of lines printed on each page.

`\setmaxlines` `\setmaxlines{`⟨*n*⟩`}`, where ⟨*n*⟩ is a positive integer ≥ 1, can be used either in the preamble or at any point of the document to set the maximum number of lines to be printed on each page. This command has the same effect as the `maxlines` option of `\SetLineation`.

```
\def\setmaxlines#1{\def\maxlines@value{#1}}
```

`\localmaxlines` Once a maximum number of lines per page has been set, `\localmaxlines{`⟨*n*⟩`}` can be used to adjust this number on a given page. As for `\setmaxlines`, ⟨*n*⟩ must be a positive integer ≥ 1.

```
\def\localmaxlines#1{%
  \luadirect{tex.sprint(ekdosis.setlocalmaxlines(\luastringN{#1}))}%
  \ignorespaces}
```

`\addtomaxlines` Unlike `\localmaxlines`, `\addtomaxlines`⟨*n*⟩ takes as argument the number of lines one wishes to add or substract from the number that has been set by `\setmaxlines`. As a result, ⟨*n*⟩ can be a positive or negative integer.

```
\def\addtomaxlines#1{%
  \luadirect{tex.sprint(ekdosis.addtomaxlines(
    \luastringO{\maxlines@value}, \luastringN{#1}))}%
  \ignorespaces}
```

`\nomaxlines` `\nomaxlines` unsets any limit previously set by `\setmaxlines`.

```
\def\nomaxlines{\luadirect{tex.sprint(ekdosis.resetlocalmaxlines())}}
```

Finally, the `\MakeLineNo` command provided by the lineno package is patched so as to trigger the insertion of `\pagebreak` when the number of lines set by `\setmaxlines` has been reached:—

```
\AddToHook{cmd/MakeLineNo/after}{%
  \ifdefined\maxlines@value
    \stepcounter{ekd@lnperpage}%
    \ifnum\value{ekd@locallnperpage} = 1
      \ifnumcomp{\theekd@lnperpage}{=}{%
        \luadirect{tex.sprint(ekdosis.getlocalmaxlines())}}{%
        \setcounter{ekd@locallnperpage}{0}%
        \setcounter{ekd@lnperpage}{0}\pagebreak}{}%
    \else
      \ifnumcomp{\theekd@lnperpage}{=}{\maxlines@value}{%
        \setcounter{ekd@lnperpage}{0}\pagebreak}{}%
    \fi
  \fi
}
```

`\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*⟩`}`.

```
\ekvdefinekeys{appnote}{
```

```
    store type = \ekdan@type,
    initial type = default
}
\NewDocumentCommand{\SetDefaultApparatus}{m}{%
  \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 commands used to insert the entries, either `\lem`, `\rdg` or `\note`:—

```
\NewDocumentCommand{\app}{O{} > { \TrimSpaces } +m}{%
  \leavevmode
  \begingroup
  \ekvset{appnote}{#1}%
  \ifekd@isinapp\ekd@appinapptrue\fi
  \ekd@isinapptrue
  \stepcounter{ekd@lab}%
  \zlabel{ekd:\theekd@lab}%
  \luadirect{ekdosis.storeabspg(
    \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}%
  \ifekd@state\add@apparatus\fi
  \luadirect{tex.sprint(ekdosis.removesp(\luastringN{#2}))}%
  \ekd@isinappfalse
  \ekd@appinappfalse
  \endgroup}
```

\App In contrast to `\app`, `\App` takes two mandatory arguments and accepts one optional argument like so: `\App[type=⟨type⟩]{⟨lemma text⟩}{⟨variants and notes⟩}`. As just described above, `type=` refers to the layer the note must go into. `\App` is strictly equivalent to `\app`, except that lemmas, variants and notes are split into two different arguments, which allows for more flexible code folding. ⟨*lemma text*⟩ is meant to receive `\lem`, while `\rdg` and `\note` go into ⟨*variants and notes*⟩.

```
\NewDocumentCommand{\App}{omm}{%
  \IfNoValueTF{#1}
  {\app{#2#3}}
  {\app[#1]{#2#3}}%
}
```

\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.

```
\NewDocumentCommand{\ekdpage}{}{%
  \luadirect{tex.sprint(ekdosis.getekdabspg())}%
}
```

`\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 reference to be inserted in the apparatus criticus.

```
\def\current@ref@arg#1#2{{%\textdir TLT%
  \unexpanded\expandafter{\ekd@refnumstyle}%
  \ifnum
    \pdf@strcmp{\getpagerefnumber{#1}}{\getpagerefnumber{#2}}
    =
    0
    \ifnum
```

```
    \pdf@strcmp{\getrefnumber{#1}}{\getrefnumber{#2}}
    =
    0
    %
    \ifekd@mapps
      \ifbool{subsq@unit@\ekdan@type}{%
          \ifnum
            \pdf@strcmp{\getrefnumber{#1}}{%
              \getrefnumber{\luadirect{tex.sprint(
                  ekdosis.getprevnotelab())}}}
            =
            0
          \else
            \LRnum{\getrefnumber{#1}}%
            \unexpanded\expandafter{\ekd@postrefnum}% issue the no
          \fi
        }%
        {\LRnum{\getrefnumber{#1}}%
          \unexpanded\expandafter{\ekd@postrefnum}}% issue the no
    \else
      \ifsubsq@unit
        %
        \ifnum
          \pdf@strcmp{\getrefnumber{#1}}{%
            \getrefnumber{\luadirect{tex.sprint(
                ekdosis.getprevnotelab())}}}
          =
          0
        \else
          \LRnum{\getrefnumber{#1}}%
          \unexpanded\expandafter{\ekd@postrefnum}% issue the no
        \fi
        %
      \else
        \LRnum{\getrefnumber{#1}}%
        \unexpanded\expandafter{\ekd@postrefnum}% issue the no
      \fi
    \fi
    %
  \else
    \ifekd@elidednumbers
      \luadirect{tex.sprint(
        ekdosis.numrange(\luastring{\getrefnumber{#1}},
        \luastring{\getrefnumber{#2}}))}%
      \unexpanded\expandafter{\ekd@postrefnum}% issue the nos
    \else
      \LRnum{\getrefnumber{#1}}--\LRnum{\getrefnumber{#2}}%
      \unexpanded\expandafter{\ekd@postrefnum}% issue the nos
    \fi
  \fi
\else
  \ifboolexpr{bool {ekd@pagelineation} or bool {ekd@pagevlineation}}
    {\LRnum{\getrefnumber{#1}}--%
      \LRnum{\getpagerefnumber{#2}}.%
      \LRnum{\getrefnumber{#2}}%
```

```
      \unexpanded\expandafter{\ekd@postrefnum}}% issue pg and ln nos
    {\LRnum{\getrefnumber{#1}}--\LRnum{\getrefnumber{#2}}%
      \unexpanded\expandafter{\ekd@postrefnum}}% issue the nos
  \fi
  \ifekdn@forcenum
    \LRnum{\getrefnumber{#1}}%
    \unexpanded\expandafter{\ekd@postrefnum}% force the no
  \fi
}}
```

`\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.

```
\def\current@ref{{%\textdir TLT%
    \unexpanded\expandafter{\ekd@refnumstyle}%
```

If the whole lemma falls on the same page...

```
    \ifnum
      \pdf@strcmp{%
        \getpagerefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}%
      {\getpagerefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}
      =
      0
```

... and on the same line,

```
      \ifnum
        \pdf@strcmp{%
          \getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-b}}%
        {\getrefnumber{\luadirect{tex.sprint(ekdosis.getlnlab())}-e}}
        =
        0
```

then, if multiple layers have been defined,

```
        \ifekd@mapps
```

and the entry is a subsequent one, when it begins on the same line as the previous one...

```
          \ifbool{subsq@unit@\ekdan@type}{%
            \ifnum
              \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(
                    ekdosis.getlnlab())}-b}}%
              {\getrefnumber{\luadirect{tex.sprint(
                    ekdosis.getprevlnlab())}-b}}
              =
              0
```

and ends likewise, then do nothing:—

```
              \ifnum
                \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(
                      ekdosis.getlnlab())}-e}}%
                {\getrefnumber{\luadirect{tex.sprint(
                      ekdosis.getprevlnlab())}-e}}
                =
                0
```

(What follows is for nested `\app` entries.)

```
              \ifekd@appinapp
                \ifnum
                  \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(
```

```
                ekdosis.getlnlab())}-b}}%
              {\getrefnumber{\luadirect{tex.sprint(
                ekdosis.getprevprevlnlab())}-b}}
              =
              0
            \else
              \LRnum{\getrefnumber{\luadirect{tex.sprint(
                ekdosis.getlnlab())}-b}}%
              \unexpanded\expandafter{\ekd@postrefnum}% issue
                                                      % the no
            \fi
          \fi
```

Otherwise, print the line number:—

```
        \else
          \LRnum{\getrefnumber{\luadirect{tex.sprint(
              ekdosis.getlnlab())}-b}}%
          \unexpanded\expandafter{\ekd@postrefnum}% issue the no
        \fi
```

If the entry begins on a new line, print the number:—

```
      \else
        \LRnum{\getrefnumber{\luadirect{tex.sprint(
            ekdosis.getlnlab())}-b}}%
        \unexpanded\expandafter{\ekd@postrefnum}% issue the no
      \fi
```

And always print the line number ahead of first entries:—

```
    }{\LRnum{\getrefnumber{\luadirect{tex.sprint(
            ekdosis.getlnlab())}-b}}%
      \unexpanded\expandafter{\ekd@postrefnum}}% issue the no
```

Now if there is only one default layer:—

```
  \else
```

On subsequent entries...

```
    \ifsubsq@unit
```

... if the whole lemma text begins on the same line as the preceding one...

```
      \ifnum
        \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(
            ekdosis.getlnlab())}-b}}%
        {\getrefnumber{\luadirect{tex.sprint(
            ekdosis.getprevlnlab())}-b}}
        =
        0
```

... and ends likewise...

```
        \ifnum
          \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(
              ekdosis.getlnlab())}-e}}%
          {\getrefnumber{\luadirect{tex.sprint(
              ekdosis.getprevlnlab())}-e}}
          =
          0
```

(What follows is for nested \app entries.)

```
        \ifekd@appinapp
          \ifnum
```

```
                \pdf@strcmp{\getrefnumber{\luadirect{tex.sprint(
                      ekdosis.getlnlab())}-b}}%
                {\getrefnumber{\luadirect{tex.sprint(
                      ekdosis.getprevprevlnlab())}-b}}
                =
                0
              \else
                \LRnum{\getrefnumber{\luadirect{tex.sprint(
                      ekdosis.getlnlab())}-b}}%
                \unexpanded\expandafter{\ekd@postrefnum}% issue the no
              \fi
            \fi
```

... then do nothing. Otherwise, print the line number:—

```
          \else
            \LRnum{\getrefnumber{\luadirect{tex.sprint(
                  ekdosis.getlnlab())}-b}}%
            \unexpanded\expandafter{\ekd@postrefnum}% issue the no
          \fi
```

If the entry begins on a new line, print the number as well:—

```
        \else
          \LRnum{\getrefnumber{\luadirect{tex.sprint(
                ekdosis.getlnlab())}-b}}%
          \unexpanded\expandafter{\ekd@postrefnum}% issue the no
        \fi
```

And always print the line number ahead of first entries:—

```
      \else
        \LRnum{\getrefnumber{\luadirect{tex.sprint(
              ekdosis.getlnlab())}-b}}%
        \unexpanded\expandafter{\ekd@postrefnum}% issue the no
      \fi
    \fi
```

Now if the lemma text breaks across lines ...

```
  \else
```

... then, depending on what has been instructed, either print the last number of a range elided:—

```
    \ifekd@elidednumbers
      \luadirect{tex.sprint(ekdosis.numrange(
        \luastring{\getrefnumber{\luadirect{tex.sprint(
              ekdosis.getlnlab())}-b}},
        \luastring{\getrefnumber{\luadirect{tex.sprint(
              ekdosis.getlnlab())}-e}}))}%
      \unexpanded\expandafter{\ekd@postrefnum}% issue the nos
```

... or in full:—

```
    \else
      \LRnum{\getrefnumber{\luadirect{tex.sprint(
            ekdosis.getlnlab())}-b}}--%
      \LRnum{\getrefnumber{\luadirect{tex.sprint(
            ekdosis.getlnlab())}-e}}%
      \unexpanded\expandafter{\ekd@postrefnum}% issue the nos
    \fi
  \fi
```

When the lemma breaks across pages:—

```
    \else
```

Print the page number and the line number when the numbering starts afresh at the top of each page:—

```
      \ifboolexpr{bool {ekd@pagelineation} or bool {ekd@pagevlineation}}
          {\LRnum{\getrefnumber{\luadirect{tex.sprint(
                  ekdosis.getlnlab())}-b}}--%
            \LRnum{\getpagerefnumber{\luadirect{tex.sprint(
                  ekdosis.getlnlab())}-e}}.%
            \LRnum{\getrefnumber{\luadirect{tex.sprint(
                  ekdosis.getlnlab())}-e}}%
            \unexpanded\expandafter{\ekd@postrefnum}}% issue pg and ln nos
```

Or just the line number if the lines are continuously numbered throughout the book:—

```
          {\LRnum{\getrefnumber{\luadirect{tex.sprint(
                  ekdosis.getlnlab())}-b}}--%
            \LRnum{\getrefnumber{\luadirect{tex.sprint(
                  ekdosis.getlnlab())}-e}}%
            \unexpanded\expandafter{\ekd@postrefnum}}% issue the nos
    \fi
```

Finally, print the number when instructed to do so:—

```
    \ifekdl@forcenum
      \LRnum{\getrefnumber{\luadirect{tex.sprint(
            ekdosis.getlnlab())}-b}}%
      \unexpanded\expandafter{\ekd@postrefnum}% force the no
    \fi
  }%
}
```

Define keys to be used by the optional arguments of `\lem` and `\rdg`:—

```
\newif\ifekdl@forcenum
\newif\ifekdl@nonum
\newif\ifekdl@nodelim
\newif\ifekdl@forcedelim
\newif\ifekdl@ilabel
% \newif\ifekdl@nowit % defined above
\ekvdefinekeys{lem}{
  code wit = \def\ekdlr@wit{#1},
  choice nowit = {true=\ekdl@nowittrue,
    false=\ekdl@nowitfalse},
  default nowit = true,
  unknown-choice nowit = \PackageError{ekdosis}{unknown
    nowit=#1}{`nowit' must be either `true' or `false'.},
  code source = \def\ekdlr@source{#1},
  code resp = \def\ekdlr@resp{#1},
  code alt  = \def\ekdlr@alt{#1},
  code pre = \def\ekdlr@pre{#1},
  code post = \def\ekdlr@post{#1},
  code prewit = \def\ekdlr@prewit{#1},
  code postwit = \def\ekdlr@postwit{#1},
  code ilabel = \ekdl@ilabeltrue\def\ilabel@val{#1},
  store type = \ekdlr@type,
  store sep = \ekdl@sep,
  noval nonum = \ekdl@nonumtrue,
  noval num = \ekdl@forcenumtrue,
```

```
  noval nodelim = \ekdl@nodelimtrue,
  noval delim = \ekdl@forcedelimtrue,
  bool nolem = \ifekdl@nolem,
  nmeta Nolem = {nodelim, nonum, nolem},
  bool nosep = \ifekdl@nosep,
  initial sep = \ekdsep
}
\ekvdefinekeys{rdg}{
  code wit = \def\ekdlr@wit{#1},
  code source = \def\ekdlr@source{#1},
  code resp = \def\ekdlr@resp{#1},
  code alt  = \def\ekdlr@alt{#1},
  code pre = \def\ekdlr@pre{#1},
  code post = \def\ekdlr@post{#1},
  code prewit = \def\ekdlr@prewit{#1},
  code postwit = \def\ekdlr@postwit{#1},
  store subsep = \ekdr@subsep,
  initial subsep = \ekdsubsep,
  bool nosubsep = \ifekdr@nosubsep,
  store type = \ekdlr@type,
  bool nordg = \ifekdr@nordg
}
```

\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.

```
\NewDocumentCommand{\rdgGrp}{O{} > {\TrimSpaces } m}{%
  \luadirect{tex.sprint(ekdosis.removesp(\luastringN{#2}))}%
}
```

\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`.

```
\def\app@lang{%
  \ifekd@mapps
  \luadirect{tex.sprint(ekdosis.getapplang(\luastring{\ekdan@type}))}%
  \else
  \ekd@singleapp@lang
  \fi
}
\def\app@note@lang{%
  \ifekd@mapps
  \luadirect{tex.sprint(ekdosis.getappnotelang(\luastring{\ekdan@type}))}%
  \else
  \ekd@singleapp@note@lang
  \fi
}
```

\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.

```
\NewDocumentCommand{\lem}{O{} m}{%
  \ifekd@appinapp
```

```
    \let\ekdlr@wit\@undefined
    \let\ekdlr@source\@undefined
    \let\ekdlr@resp\@undefined
    \let\ekdlr@alt\@undefined
    \let\ekdlr@pre\@undefined
    \let\ekdlr@post\@undefined
    \let\ekdlr@prewit\@undefined
    \let\ekdlr@postwit\@undefined
  \fi
  \ekd@isinlemtrue
  \bgroup
  \ekdl@forcenumfalse
  \ekdl@nonumfalse
  \ekdl@ilabelfalse
  \ifekd@mapps
    \luadirect{tex.sprint(ekdosis.setapp_neg_pos(\luastringO{\ekdan@type}))}%
  \fi
  \ekvset{lem}{#1}%
  \ifekdl@ilabel
  \luadirect{ekdosis.dolnlab(\luastringN{#2},
    \luastringO{\ilabel@val})}%
  \else
  \luadirect{ekdosis.dolnlab(\luastringN{#2})}%
  \fi
  \null
  \ekd@test@lang
  \ifekd@mapps
    \ifnum
    \luadirect{tex.sprint(ekdosis.get_bagunits(\luastringO{\ekdan@type}))}
    = 1
    \boolfalse{subsq@unit@\ekdan@type}%
    \fi
    \luadirect{ekdosis.increment_bagunits(\luastringO{\ekdan@type})}%
    \def\ekd@munit@delim{%
      \luadirect{tex.sprint(ekdosis.getappdelim(\luastringO{\ekdan@type}))}}%
    \luadirect{tex.sprint(ekdosis.limit_bagunits(\luastringO{\ekdan@type}))}%
  \fi%
  \ifekdl@nolem
    \edef\lem@app{%
      % \hskip .75em
      \ifekd@mapps
        \unless\ifekdl@nodelim
          \ifbool{subsq@unit@\ekdan@type}%
            {\ekd@munit@delim}{\ifekdl@forcedelim\ekd@munit@delim\fi}%
        \fi
      \else
        \unless\ifekdl@nodelim
          \ifsubsq@unit
            \unexpanded\expandafter{\ekd@unit@delim}%
          \else
            \ifekdl@forcedelim\unexpanded\expandafter{\ekd@unit@delim}\fi
          \fi
        \fi
      \fi%
      \unless\ifekdl@nonum\current@ref\fi
```

```
      \relax
    }%\hskip .25em}%
  \else
  \ifbool{al@rlmode}{%
    \edef\lem@app{%
      % \hskip .75em
      \ifekd@mapps
        \unless\ifekdl@nodelim
          \ifbool{subsq@unit@\ekdan@type}%
            {\ekd@munit@delim}{\ifekdl@forcedelim\ekd@munit@delim\fi}%
        \fi
      \else
        \unless\ifekdl@nodelim
          \ifsubsq@unit
            \unexpanded\expandafter{\ekd@unit@delim}%
          \else
            \ifekdl@forcedelim\unexpanded\expandafter{\ekd@unit@delim}\fi
          \fi
        \fi
      \fi
      \unless\ifekdl@nonum\current@ref\fi%\hskip .25em
      \ifdefined\ekdlr@alt%
        \ifrtl@app
          \ifdefined\ekdlr@post%
            \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
          {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
            \unexpanded\expandafter{\ekdlr@alt}}%
          \ifdefined\ekdlr@pre%
            \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
        \else
          \ifdefined\ekdlr@pre%
            \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
          {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
            \unexpanded\expandafter{\ekdlr@alt}}%
          \ifdefined\ekdlr@post%
            \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
        \fi
      \else
        \ifrtl@app
          \ifdefined\ekdlr@post%
            \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
          {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
            \unexpanded{#2}}%
          \ifdefined\ekdlr@pre%
            \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
        \else
          \ifdefined\ekdlr@pre%
            \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
          {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
            \unexpanded{#2}}%
          \ifdefined\ekdlr@post%
            \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
        \fi
      \fi
      \ifrtl@app
```

```
      \ifdefined\ekdlr@postwit%
        \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
      \unless\ifekdl@nowit
        \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
      \fi
      \ifdefined\ekdlr@prewit%
        \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
      \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
      \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
    \else
      \ifdefined\ekdlr@prewit%
        \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
      \unless\ifekdl@nowit
        \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
      \fi
      \ifdefined\ekdlr@postwit%
        \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
      \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
      \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
    \fi
    \ifekdl@nosep\else\unexpanded\expandafter{\ekdl@sep}\fi
  }%
}%
{%
  \edef\lem@app{%
    % \hskip .75em
    \ifekd@mapps
      \unless\ifekdl@nodelim
        \ifbool{subsq@unit@\ekdan@type}%
          {\ekd@munit@delim}{\ifekdl@forcedelim\ekd@munit@delim\fi}%
      \fi
    \else
      \unless\ifekdl@nodelim
        \ifsubsq@unit
          \unexpanded\expandafter{\ekd@unit@delim}%
        \else
          \ifekdl@forcedelim\unexpanded\expandafter{\ekd@unit@delim}\fi
        \fi
      \fi
    \fi%
    \unless\ifekdl@nonum\current@ref\fi%\hskip .25em
    \ifdefined\ekdlr@alt%
      \ifdefined\ekdlr@pre%
        \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
      \ifbool{ekd@lang@pkg}%
      {{\unexpanded\expandafter{\ekd@lemmastyle}%
          \noexpand\selectlanguage{\app@lang}%
          \unexpanded\expandafter{\ekdlr@alt}}}%
      {\unexpanded\expandafter{\ekd@lemmastyle}%
        \unexpanded\expandafter{\ekdlr@alt}}%
      \ifdefined\ekdlr@post%
        \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
    \else
      \ifdefined\ekdlr@pre%
        \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
```

```
          \ifbool{ekd@lang@pkg}%
          {{\unexpanded\expandafter{\ekd@lemmastyle}%
              \noexpand\selectlanguage{\app@lang}%
              \unexpanded{#2}}}{%
            {\unexpanded\expandafter{\ekd@lemmastyle}\unexpanded{#2}}}%
        \ifdefined\ekdlr@post%
          \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
      \fi
      \ifdefined\ekdlr@prewit%
        \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
      \unless\ifekdl@nowit
        \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
      \fi
      \ifdefined\ekdlr@postwit%
        \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
      \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
      \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
      \ifekdl@nosep\else\unexpanded\expandafter{\ekdl@sep}\fi
    }%
  }%
  \fi
  \ifekd@mapps
    \ifekdl@ilabel
      \append@ln@app[\ekdan@type][\ilabel@val]{\lem@app}%
    \else
      \append@ln@app[\ekdan@type]{\lem@app}%
    \fi
  \else
    \ifekdl@ilabel
      \append@ln@app[][\ilabel@val]{\lem@app}
    \else
      \append@ln@app{\lem@app}%
    \fi
  \fi%
  \egroup%
  \ekd@isinlemfalse%
  \subsq@unittrue%
}
```

\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@subsq@rdg` to true, which instructs ekdosis that "subseparators" may be used for subsequent entries.

```
\newif\ifekd@subsq@rdg
\NewDocumentCommand{\rdg}{O{} m}{%
  \bgroup%
  \ekvset{rdg}{#1}%
  \ekd@test@lang
  % \ifekdr@nordg\append@app{}\else% do we need \append@app{} here? If
  %                               % so, keep in mind \ifekd@mapps,
  %                               like so:
  \unless\ifekdr@nordg
  \ifbool{al@rlmode}{%
    \edef\rdg@app{%
      \ifekd@subsq@rdg
```

```
      \unless\ifekdr@nosubsep\unexpanded\expandafter{\ekdr@subsep}\fi
    \fi
    \ifdefined\ekdlr@alt%
      \ifrtl@app
        \ifdefined\ekdlr@post%
          \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
        {\textdir TRT\unexpanded\expandafter{\ekd@readingstyle}%
          \unexpanded\expandafter{\ekdlr@alt}}%
        \ifdefined\ekdlr@pre%
          \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
      \else
        \ifdefined\ekdlr@pre%
          \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
        {\textdir TRT\unexpanded\expandafter{\ekd@readingstyle}%
          \unexpanded\expandafter{\ekdlr@alt}}%
        \ifdefined\ekdlr@post%
          \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
      \fi
    \else
      \ifrtl@app
        \ifdefined\ekdlr@post%
          \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
        {\textdir TRT\unexpanded\expandafter{\ekd@readingstyle}%
          \unexpanded{#2}}%
        \ifdefined\ekdlr@pre%
          \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
      \else
        \ifdefined\ekdlr@pre%
          \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
        {\textdir TRT\unexpanded\expandafter{\ekd@readingstyle}%
          \unexpanded{#2}}%
        \ifdefined\ekdlr@post%
          \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
      \fi
    \fi
    \ifrtl@app
      \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
      \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
      \ifdefined\ekdlr@postwit%
        \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
      \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
      \ifdefined\ekdlr@prewit%
        \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
    \else
      \ifdefined\ekdlr@prewit%
        \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
      \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
      \ifdefined\ekdlr@postwit%
        \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
      \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
      \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
    \fi
  }%
}%
{%
```

```
    \edef\rdg@app{%
      \ifekd@subsq@rdg
        \unless\ifekdr@nosubsep\unexpanded\expandafter{\ekdr@subsep}\fi
      \fi
      \ifdefined\ekdlr@alt%
        \ifdefined\ekdlr@pre%
          \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
        \ifbool{ekd@lang@pkg}%
        {{\unexpanded\expandafter{\ekd@readingstyle}%
            \noexpand\selectlanguage{\app@lang}%
            \unexpanded\expandafter{\ekdlr@alt}}}%
        {\unexpanded\expandafter{\ekd@readingstyle}%
          \unexpanded\expandafter{\ekdlr@alt}}%
        \ifdefined\ekdlr@post%
          \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
      \else
        \ifdefined\ekdlr@pre%
          \space\unexpanded\expandafter{\ekdlr@pre}\space\else\fi
          \ifbool{ekd@lang@pkg}%
          {{\unexpanded\expandafter{\ekd@readingstyle}%
              \noexpand\selectlanguage{\app@lang}\unexpanded{#2}}}{%
            {\unexpanded\expandafter{\ekd@readingstyle}\unexpanded{#2}}}%
        \ifdefined\ekdlr@post%
          \space\unexpanded\expandafter{\ekdlr@post}\space\else\fi
      \fi
      \ifdefined\ekdlr@prewit%
        \space\unexpanded\expandafter{\ekdlr@prewit}\space\else\fi
      \ifdefined\ekdlr@wit\space\getsiglum{\ekdlr@wit}\else\fi
      \ifdefined\ekdlr@postwit%
        \space\unexpanded\expandafter{\ekdlr@postwit}\else\fi
      \ifdefined\ekdlr@source\space\getsiglum{\ekdlr@source}\else\fi
      \ifdefined\ekdlr@resp\space\getsiglum{\ekdlr@resp}\else\fi
    }%
  }%
  \ifekd@mapps
    \append@app[\ekdan@type]{\rdg@app}%
  \else
    \append@app{\rdg@app}%
  \fi
\fi
\egroup
\ekd@subsq@rdgtrue
}
```

Define keys to be used by the optional argument of `\note` when this command is found outside `\app`:—

```
\newif\ifekdn@forcenum
\ekvdefinekeys{note}{
  store type = \ekdan@type,
  store lem = \ekdn@lem,
  code labelb = \def\ekdn@labelb{#1},
  code labele = \def\ekdn@labele{#1},
  bool nodelim = \ifekdn@nodelim,
  store sep = \ekdn@sep,
  bool nosep = \ifekdn@nosep,
  initial type = default,
```

```
  initial sep = \ekdsep,
  bool nonum = \ifekdn@nonum,
  noval num = \ekdn@forcenumtrue
}
```

`\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.

```
\NewDocumentCommand{\note@noapp}{O{} +m}{%
  \leavevmode
  \bgroup%
  \ekvset{note}{#1}%
  \ekd@test@lang
  \stepcounter{ekd@lab}%
  \zlabel{ekd:\theekd@lab}%
  \luadirect{ekdosis.storeabspg(
    \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}%
  \ifekd@state\add@apparatus\fi%
  \ifekd@mapps%
    \ifnum%
      \luadirect{tex.sprint(ekdosis.get_bagunits(\luastringO{\ekdan@type}))}
      = 1
      \boolfalse{subsq@unit@\ekdan@type}%
    \fi%
  \luadirect{ekdosis.increment_bagunits(\luastringO{\ekdan@type})}%
    \def\ekd@munit@delim{%
      \luadirect{tex.sprint(ekdosis.getappdelim(\luastringO{\ekdan@type}))}}%
    \luadirect{tex.sprint(ekdosis.limit_bagunits(\luastringO{\ekdan@type}))}%
  \fi%
  \ifdefined\ekdn@labelb%
    \luadirect{tex.sprint(ekdosis.setnotelab(\luastringO{\ekdn@labelb}))}%
    \ifdefined\ekdn@labele\else\def\ekdn@labele{\ekdn@labelb}\fi%
  \else\PackageError{ekdosis}{missing labelb}{`labelb' must be
    set.}\fi%
  \ifbool{al@rlmode}%
  {\edef\note@contents{%
      % \hskip .75em
      \ifekd@mapps
      \unless\ifekdn@nodelim
        \ifbool{subsq@unit@\ekdan@type}%
               {\ekd@munit@delim}{}%
      \fi
      \else
        \unless\ifekdn@nodelim
          \ifsubsq@unit\unexpanded\expandafter{\ekd@unit@delim}\fi
        \fi
      \fi%
      \unless\ifekdn@nonum\current@ref@arg{\ekdn@labelb}{\ekdn@labele}\fi%\hskip .25em
      \ifdefined\ekdn@lem%
      {\textdir TRT\unexpanded\expandafter{\ekd@lemmastyle}%
        \unexpanded\expandafter{\ekdn@lem}}%
          \unless\ifekdn@nosep
          \unexpanded\expandafter{\ekdn@sep}\fi
          \else\fi%
```

```
            {\textdir TRT\unexpanded{#2}}}}%
  {\edef\note@contents{%
      % \hskip .75em
      \ifekd@mapps
        \unless\ifekdn@nodelim
          \ifbool{subsq@unit@\ekdan@type}%
                 {\ekd@munit@delim}{}%
        \fi
      \else
        \unless\ifekdn@nodelim
          \ifsubsq@unit\unexpanded\expandafter{\ekd@unit@delim}\fi
        \fi
      \fi%
      \unless\ifekdn@nonum\current@ref@arg{\ekdn@labelb}{\ekdn@labele}\fi%\hskip .25em
      \ifdefined\ekdn@lem
        \ifbool{ekd@lang@pkg}%
        {{\unexpanded\expandafter{\ekd@lemmastyle}%
            \noexpand\selectlanguage{\app@lang}%
              \unexpanded\expandafter{\ekdn@lem}}}%
          {\unexpanded\expandafter{\ekd@lemmastyle}%
            \unexpanded\expandafter{\ekdn@lem}}%
          \unless\ifekdn@nosep
          \unexpanded\expandafter{\ekdn@sep}\fi
          \else\fi%
          \ifbool{ekd@lang@pkg}%
          {{\noexpand\selectlanguage{\app@note@lang}\unexpanded{#2}}}{%
            {\unexpanded{#2}}}}}%
  \ifekd@mapps
    \unconditional@appin[\ekdan@type]{\note@contents}%
  \else
    \unconditional@appin{\note@contents}%
  \fi
    \luadirect{ekdosis.setprevnotelab(\luastringO{\ekdn@labelb})}%
  \egroup
  \subsq@unittrue
  \ignorespaces
}
```

Define keys to be used by the optional argument of `\note` when this command is found inside `\app`:—

```
\ekvdefinekeys{ekd@note}{
  store pre = \pre@value,
  store post = \post@value,
  nmeta sep = {post=\ekdsep},
  nmeta subsep = {pre=\ekdsubsep}
}
```

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.

```
\NewDocumentCommand{\ekd@note}{O{} m}{%
  \bgroup
  \ekvset{ekd@note}{#1}%
  \edef\note@contents{%
```

```
    \ekvifdefinedNoVal{ekd@note}{pre}{}{%
      \unexpanded\expandafter{\pre@value}}%
    \unexpanded{#2}%
    \ekvifdefinedNoVal{ekd@note}{post}{}{%
      \unexpanded\expandafter{\post@value}}%
  }%
  \ifekd@mapps%
  \append@app[\ekdan@type]{\note@contents}%
  \else%
  \append@app{\note@contents}%
  \fi%
  \egroup
}
\NewDocumentCommand{\ekd@note@star}{O{} m}{%
  \bgroup
  \ekvset{ekd@note}{#1}%
  \edef\note@contents{%
    \ekvifdefinedNoVal{ekd@note}{pre}{}{%
      \unexpanded\expandafter{\pre@value}}%
    \if@pkg@parnotes
      \unskip\noexpand\parnote{\unexpanded{#2}}%
    \else
      \unskip\noexpand\footnote{\unexpanded{#2}}%
    \fi
    \ekvifdefinedNoVal{ekd@note}{post}{}{%
      \unexpanded\expandafter{\post@value}}%
  }%
  \ifekd@mapps
    \append@app[\ekdan@type]{\note@contents}%
  \else
    \append@app{\note@contents}%
  \fi
  \egroup
}
\NewDocumentCommand{\note@app}{s O{} +m}{%
  \ifbool{al@rlmode}{%
    \IfBooleanTF{#1}{\ekd@note@star[#2]{%
          {\textdir TRT#3}}}
    {\ekd@note[#2]{{\textdir TRT#3}}}%
  }{%
    \IfBooleanTF{#1}{\ekd@note@star[#2]{#3}}
    {\ekd@note[#2]{#3}}%
  }%
}
```

`\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:—

```
\NewDocumentCommand{\note}{s O{} +m}{%
  \ifekd@state%
    \ifekd@isinapp%
      \ifekd@isinlem%
        \note@noapp[#2]{#3}%
      \else%
        \IfBooleanTF{#1}{\note@app*[#2]{#3}}{\note@app[#2]{#3}}%
```

```
      \fi%
    \else%
      \note@noapp[#2]{#3}%
      \fi%
  \fi%
}
```

Combining Footnotes Into a Single Paragraph Footnotes combined as one paragraph are inserted with the standard `\footnote` command which is redirected to `\unconditional@appin` when `ekd@parafootnotes` is set to `true`:—

```
\newif\ifekd@parafootnotes
```

The keys to be used by `\SetFootnotes` are defined below:—

```
\ekvdefinekeys{ekd@footnotes}{
  choice arrangement = {
    column = \unless\ifekd@alignmentstarted
             \pcol@fnlayout@c\fi,
    page = \unless\ifekd@alignmentstarted
             \pcol@fnlayout@p\fi,
    merge = \unless\ifekd@alignmentstarted
              \pcol@fnlayout@m\fi},
  unknown-choice arrangement = \PackageError{ekdosis}{unknown
    arrangement=#1}{`arrangement' must be either `column',
    `page' or `merge'.},
  noval reset = \ekd@parafootnotesfalse
                \unless\ifekd@alignmentstarted
                \pcol@fnlayout@c\fi,
  bool paragraph = \ifekd@parafootnotes,
  store type = \ekdfn@type,
  initial type = default,
  code textfnmark = \protected\def\ekd@around@fnmark##1{#1},
  initial textfnmark = \textsuperscript{#1},
  code appfnmark = \protected\def\ekdappfnmark##1{#1},
  initial appfnmark = \textsuperscript{#1}
}
```

A conditional `\ifekd@chfnmark` is defined so that ekdosis may know whether the footnote number has to be replaced with some other mark.

```
\newif\ifekd@chfnmark
```

These are the keys to be used by `\ekd@parafootnote`:—

```
\ekvdefinekeys{ekd@infootnotes}{
  store type = \ekdfn@type,
  code mark = \ekd@chfnmarktrue
              \protected\def\ekd@fnmark@value{#1}
}
```

`\SetFootnotes` `\SetFootnotes{`⟨*options*⟩`}` can be used in the preamble or at any point of the document. `layout=paragraph` instructs ekdosis to combine all footnotes into a single paragraph. Then the other `key-value` options that are defined just above allow to format the marks and specify the layer where the notes are to be printed.

```
\ekvsetdef\SetFootnotes{ekd@footnotes}
```

`\ekd@parafootnote` `\ekd@parafootnote[`⟨*options*⟩`]{`⟨*contents*⟩`}` holds the new definition of `\footnote`. This command accepts an optional argument that can be used to specify the layer in which the

contents of the footnote is to be printed or any other mark to be printed in place of the footnote number.

```
\NewDocumentCommand{\ekd@parafootnote}{O{} m}{%
  \ifekd@state
    \bgroup
    \ekvset{ekd@infootnotes}{#1}%
    \ifekd@chfnmark
      \edef\ekd@parafn@mark{%
        \LRnum{\unexpanded\expandafter{\ekd@fnmark@value}}}%
      \edef\ekd@inparafn@mark{%
        \ekdappfnmark{%
          \LRnum{\unexpanded\expandafter{\ekd@fnmark@value}}}}%
    \else
      \refstepcounter{footnote}%
      \edef\ekd@parafn@mark{\LRnum{\thefootnote}}%
      \edef\ekd@inparafn@mark{\ekdappfnmark{\LRnum{\thefootnote}}}%
    \fi
    \leavevmode
    \stepcounter{ekd@lab}%
    \zlabel{ekd:\theekd@lab}%
    \luadirect{ekdosis.storeabspg(
      \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}%
    \add@apparatus
    \luadirect{ekdosis.increment_bagunits(\luastringO{\ekdfn@type})}%
    \luadirect{tex.sprint(
      ekdosis.limit_bagunits(\luastringO{\ekdan@type}))}%
    \ekd@around@fnmark{\ekd@parafn@mark}%
    \ifekd@mapps
      \unconditional@appin[\ekdfn@type]{\ekd@inparafn@mark #2}%
    \else
      \unconditional@appin{\ekd@inparafn@mark #2}%
    \fi
    \egroup
  \else
    \footnote{#2}%
  \fi
}
```

\ekd@parafootnotemark `\ekd@parafootnotemark[`⟨*mark*⟩`]` replaces the definition of the standard `\footnotemark` command.

```
\NewDocumentCommand{\ekd@parafootnotemark}{o}{%
  \ifekd@state
    \bgroup
    \IfNoValueTF{#1}{%
      \refstepcounter{footnote}%
      \edef\ekd@parafn@mark{\LRnum{\thefootnote}}%
    }{%
      \edef\ekd@parafn@mark{%
        \LRnum{\unexpanded\expandafter{#1}}}%
    }%
    \leavevmode
    \ekd@around@fnmark{\ekd@parafn@mark}%
    \egroup
  \else
    \IfNoValueTF{#1}{\footnotemark}{\footnotemark[#1]}%
```

```
  \fi
}
```

\ekd@parafootnotetext `\ekd@parafootnotetext[`⟨*options*⟩`]{`⟨*text*⟩`}` replaces the definition of the standard `\footnotetext` command. It does the same as `\ekd@parafootnote` except that it does not print any mark in the main text.

```
\NewDocumentCommand{\ekd@parafootnotetext}{O{} m}{%
  \ifekd@state
    \bgroup
    \ekvset{ekd@infootnotes}{#1}%
    \ifekd@chfnmark
      \edef\ekd@parafn@mark{%
        \LRnum{\unexpanded\expandafter{\ekd@fnmark@value}}}%
      \edef\ekd@inparafn@mark{%
        \ekdappfnmark{%
          \LRnum{\unexpanded\expandafter{\ekd@fnmark@value}}}}%
    \else
      \edef\ekd@parafn@mark{\LRnum{\thefootnote}}%
      \edef\ekd@inparafn@mark{\ekdappfnmark{\LRnum{\thefootnote}}}%
    \fi
    \leavevmode
    \stepcounter{ekd@lab}%
    \zlabel{ekd:\theekd@lab}%
    \luadirect{ekdosis.storeabspg(
      \luastring{\zref@extract{ekd:\theekd@lab}{abspage}})}%
    \add@apparatus
    \luadirect{ekdosis.increment_bagunits(\luastringO{\ekdfn@type})}%
    \luadirect{tex.sprint(
      ekdosis.limit_bagunits(\luastringO{\ekdan@type}))}%
    \ifekd@mapps
      \unconditional@appin[\ekdfn@type]{\ekd@inparafn@mark #2}%
    \else
      \unconditional@appin{\ekd@inparafn@mark #2}%
    \fi
    \egroup
  \else
    \footnotetext[#1]{#2}%
  \fi
}
```

Emendations and Conjectures Here follows the `key-value` options to be used by `\SetCritSymbols` below:—

```
\ekvdefinekeys{ekd@corr}{
  store suppbegin = \suppb@value,
  store suppend = \suppe@value,
  store delbegin = \delb@value,
  store delend = \dele@value,
  store sicbegin = \sicb@value,
  store sicend = \sice@value,
  store gapmark = \gapm@value,
  initial suppbegin = \ifbool{al@rlmode}{>}{<},
  initial suppend = \ifbool{al@rlmode}{<}{>},
  initial delbegin = \ifbool{al@rlmode}{\}}{\{},
  initial delend = \ifbool{al@rlmode}{\{}{\}},
  initial sicbegin = \dag,
```

```
  initial sicend = \dag,
  initial gapmark = ***,
  bool keepinapp = \ifekd@keepinapp
}
```

\supplied `\supplied{`⟨*text*⟩`}` takes as mandatory argument the text added or supplied by conjecture.

```
\NewDocumentCommand{\supplied}{m}{%
  \ifekd@inside@app
    \ifekd@keepinapp
      \suppb@value #1\suppe@value
    \else
      #1%
    \fi
  \else
    \suppb@value #1\suppe@value
  \fi
}
```

\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.

```
\NewDocumentCommand{\surplus}{m}{%
  \ifekd@inside@app
    \ifekd@keepinapp
      \delb@value #1\dele@value
    \else
      #1%
    \fi
  \else
    \delb@value #1\dele@value
  \fi
}
```

\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*⟩.

```
\NewDocumentCommand{\sic}{s m}{%
  \ifekd@inside@app
    \ifekd@keepinapp
      \IfBooleanTF{#1}
        {\sicb@value #2}
        {\sicb@value #2\sice@value}%
    \else
      #2%
    \fi
  \else
    \IfBooleanTF{#1}
      {\sicb@value #2}
      {\sicb@value #2\sice@value}%
  \fi
}
```

\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.

```
\NewDocumentCommand{\gap}{m}{%
  \gapm@value
}
```

\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.

```
\NewDocumentCommand{\SetCritSymbols}{m}{
  \ekvset{ekd@corr}{#1}
}
```

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.

```
\NewDocumentCommand{\ilabel}{m}{%
  \luadirect{tex.sprint(ekdosis.getindexedlab(\luastringN{#1}))}%
}
```

\lacunaStart
\lacunaEnd

```
\NewDocumentCommand{\lacunaStart}{O{}}{\ignorespaces}
\NewDocumentCommand{\lacunaEnd}{O{}}{\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.

```
\NewDocumentCommand{\apparatus}{}{%
  \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.

```
\NewDocumentCommand{\test@apparatus}{}{%
  \luadirect{tex.sprint(ekdosis.testapparatus())}}
\NewDocumentCommand{\ekd@storecol}{}{%
  \luadirect{ekdosis.storecurcol(\luastring{\thecolumn})}%
}
```

Start and stop ekdosis:

```
\NewDocumentCommand{\EkdosisOn}{}{%
  \ekd@statetrue
  \ifekd@parafootnotes
    \RenewCommandCopy\footnote\ekd@parafootnote
    \RenewCommandCopy\footnotemark\ekd@parafootnotemark
    \RenewCommandCopy\footnotetext\ekd@parafootnotetext
  \fi
}
\NewDocumentCommand{\EkdosisOff}{}{%
  \ekd@statefalse%
}
```

Neutralize unwanted commands provided by lineno within the ekdosis environment:—

```
\def\ekd@setlineno{%
  \let\setpagewiselinenumbers\relax%
  \let\pagewiselinenumbers\relax%
```

```
  \let\endpagewiselinenumbers\relax%
  \let\runningpagewiselinenumbers\relax%
  \let\realpagewiselinenumbers\relax%
}
```

`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.

```
\NewDocumentEnvironment{ekdosis}{+b}{%
  \ekd@setlineno
  \ifekd@hidelinenumbers
    \def\thelinenumber{}%
  \fi
  \runninglinenumbers
    \EkdosisOn#1}{%
    \EkdosisOff
  \endrunninglinenumbers
  \iftei@export
  \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:—

```
\ekvdefinekeys{ekd@align}{
  store tcols = \tcols@num,
  store lcols = \lcols@num,
  store texts = \texts@value,
  store apparatus = \apparatus@value,
  bool paired = \ifekd@paired,
    choice lineation = {page = \ekd@pagelineationtrue
                               \ekd@hidelinenumbersfalse,
    document = \ekd@pagelineationfalse
               \ekd@hidelinenumbersfalse,
    none = \ekd@pagelineationtrue
           \ekd@hidelinenumberstrue},
  unknown-choice lineation = \PackageError{ekdosis}{unknown
    lineation=#1}{`lineation' must be either `page', `document' or
    `none'.},
  choice segmentation = {auto = \def\segmentation@val{auto},
                         noauto = \def\segmentation@val{noauto}},
  unknown-choice segmentation = \PackageError{ekdosis}{unknown
    segmentation=#1}{`segmentation' must be either `auto' or
    `noauto'.},
  bool flush = \ifekd@flushapp,
  initial tcols = 2,
  initial lcols = 1,
  initial texts = edition;translation,
  initial apparatus = edition,
  default segmentation = auto
}
```

`\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.

```
\NewDocumentCommand{\SetAlignment}{m}{
```

```
  \ekvset{ekd@align}{#1}
}
```

Patch paracol to insert a hook in `\pcol@nextpage`. This hook is used to reset line numbers on new pages.

```
\patchcmd{\pcol@nextpage}{%
  \endgroup}{%
  \ifekd@pagelineation\resetlinenumber\fi
  \endgroup}{}{}
```

`\EkdosisColStart` and `\EkdosisColStop` initialize columns meant to receive edition texts. These commands are used internally by ekdosis.

```
\NewDocumentCommand{\EkdosisColStart}{}{%
  \ekd@setlineno
  \runninglinenumbers
  \ekd@storecol
  \stepcounter{ekd@lab}%
  \zlabel{ekd:\theekd@lab}%
  \luadirect{%
    ekdosis.storeabspg(\luastring{\zref@extract{ekd:\theekd@lab}{abspage}},
    "pg_i")}%
  \ifekd@pagelineation
    \luadirect{tex.sprint(ekdosis.checkresetlineno())}
  \fi
  \ifekd@hidelinenumbers
    \def\thelinenumber{}%
  \fi
  \ifekd@parafootnotes
    \RenewCommandCopy\footnote\ekd@parafootnote
    \RenewCommandCopy\footnotemark\ekd@parafootnotemark
    \RenewCommandCopy\footnotetext\ekd@parafootnotetext
  \fi
}
\NewDocumentCommand{\EkdosisColStop}{}{%
  \stepcounter{ekd@lab}%
  \zlabel{ekd:\theekd@lab}%
  \luadirect{%
    ekdosis.storeabspg(\luastring{\zref@extract{ekd:\theekd@lab}{abspage}},
    "pg_ii")}%
  \endrunninglinenumbers
}
```

`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".

```
\newif\ifekd@alignmentstarted
\NewDocumentEnvironment{alignment}{O{}}
{%
  \ekd@alignmentstartedtrue
  \ekvset{ekd@align}{#1}%
  \luadirect{ekdosis.mkenvdata(
```

```
    \luastring{\texts@value},
    "texts"
    )}
  \ifekd@flushapp
    \luadirect{ekdosis.newalignment("set")}
  \fi
  \luadirect{ekdosis.mkenvdata(
    \luastring{\apparatus@value}, "apparatus"
    )}
  \setrunninglinenumbers
  \luadirect{tex.sprint(ekdosis.mkenv())}
  \ifekd@paired
    \begin{paracol}[\lcols@num]{\tcols@num}
    \else
      \begin{paracol}[\lcols@num]*{\tcols@num}
    \fi
    }
  {\if@pkg@breakable\flushpage\fi
  \end{paracol}
  \iftei@export\luadirect{ekdosis.export_coldata_totei()}\fi
  \ifekd@flushapp
    \luadirect{ekdosis.newalignment("reset")}
  \fi
  \luadirect{ekdosis.flushenvdata()}
  \luadirect{ekdosis.flushcolnums()}
  }
```

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.

```
\NewDocumentCommand{\ekd@storemark}{m}{%
  \stepcounter{ekd@lab}%
  \label{ekd:\theekd@lab}%
  \luadirect{ekdosis.storehfmark(
    \luastring{\getpagerefnumber{ekd:\theekd@lab}},
    \luastringN{#1})}%
}
```

\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.

```
\NewDocumentCommand{\endmark}{}{%
  \stepcounter{ekd@lab}%
  \label{ekd:\theekd@lab}%
  \luadirect{ekdosis.storehfmark(
    \luastring{\getpagerefnumber{ekd:\theekd@lab}},
    "", "endmk")}%
  \ifdefined\xspace\xspace\fi
}
```

\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.

```
\NewDocumentCommand{\ekdmark}{}{%
  \luadirect{tex.sprint(ekdosis.gethfmark(\luastring{\thepage}))}%
}
```

\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 layed out on "paired" facing pages as described above on page 30, the second letter, `E` or `O`, must obviously be omitted for headers and footers are the same on every facing page:—

```
\ekvdefinekeys{ekd@marks}{
  choice mark = {HEL = \def\ekd@mk{HEL},
    HEC = \def\ekd@mk{HEC},
    HER = \def\ekd@mk{HER},
    HOL = \def\ekd@mk{HOL},
    HOC = \def\ekd@mk{HOC},
    HOR = \def\ekd@mk{HOR},
    FEL = \def\ekd@mk{FEL},
    FEC = \def\ekd@mk{FEC},
    FEL = \def\ekd@mk{FER},
    FOL = \def\ekd@mk{FOL},
    FOC = \def\ekd@mk{FOC},
    FOL = \def\ekd@mk{FOR}},
  unknown-choice mark = \PackageError{ekdosis}{unknown mark=#1}{`mark'
    must be either `HEL', `HEC', `HER', `HOL', `HOC', `HOR', `FEL',
    \MessageBreak `FEC', `FER', `FOL', `FOC' or `FOR'.}
}
\NewDocumentCommand{\ekd@printmark}{m m}{%
  \bgroup
  \ekvset{ekd@marks}{mark = #1}%
  \luadirect{tex.sprint(ekdosis.printmark(\luastringN{#2},
    \luastringO{\ekd@mk}))}%
  \egroup
}
\NewDocumentCommand{\ekdprintmark}{m m}{%
  \def\@tempa{#1}%
  \def\mk@HL{HL}\def\mk@HC{HC}\def\mk@HR{HR}%
  \def\mk@FL{FL}\def\mk@FC{FC}\def\mk@FR{FR}%
  \ifx\@tempa\mk@HL
    \csname ekd@printmark\endcsname{HEL}{\csname
      ekd@printmark\endcsname{HOL}{#2}}%
  \else
  \ifx\@tempa\mk@HC
    \csname ekd@printmark\endcsname{HEC}{\csname
      ekd@printmark\endcsname{HOC}{#2}}%
  \else
  \ifx\@tempa\mk@HR
    \csname ekd@printmark\endcsname{HER}{\csname
      ekd@printmark\endcsname{HOR}{#2}}%
  \else
  \ifx\@tempa\mk@FL
```

```
    \csname ekd@printmark\endcsname{FEL}{\csname
      ekd@printmark\endcsname{FOL}{#2}}%
  \else
  \ifx\@tempa\mk@FC
    \csname ekd@printmark\endcsname{FEC}{\csname
      ekd@printmark\endcsname{FOC}{#2}}%
  \else
  \ifx\@tempa\mk@FR
    \csname ekd@printmark\endcsname{FER}{\csname
      ekd@printmark\endcsname{FOR}{#2}}%
  \else
    \csname ekd@printmark\endcsname{#1}{#2}%
  \fi\fi\fi\fi\fi
}
```

`\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:—

```
\NewDocumentCommand{\ekdEOprint}{m m}{%
  \ifnumodd{\theabspage}{#1}{#2}%
}
```

`\ekdnofhmarks` Once the signposts are marked with `\ekdprintmark`, `\ekdnohfmarks` has the same effect as the LaTeX standard command `\thispagestyle{empty}`.

```
\NewDocumentCommand{\ekdnohfmarks}{}{%
  \luadirect{ekdosis.nohfmark()}%
}
```

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

```
\NewDocumentCommand{\ekdresethfmarks}{}{%
  \luadirect{ekdosis.resethfmark()}%
}
```

"Mirrored" paired pages An easy way to have mirrored paired pages is to use a dedicaded 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:—

```
\newcounter{pairedpage}
\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.

```
\NewDocumentCommand{\setpairedpagenum}{m}{%
```

```
  \setcounter{pairedpage}{\number\numexpr#1-1}%
}
\NewDocumentCommand{\setpairedpage}{}{%
  \ifnumodd{\thepage}{\setcounter{page}{\thepairedpage}}
  {\stepcounter{pairedpage}\setcounter{page}{\thepairedpage}}%
}
\def\resetpagenumber{%
  \ifnumodd{\thepairedpage}{}{\addtocounter{page}{-1}}
}
```

Divisions of the Body ekdosis can convert `\book`, `\part`, `\chapter`, `\section`, `\subsection` and `\subsubsection` into corresponding TEI 'numbered' `<div`$_n$`>` 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.

```
\NewDocumentCommand{\MkBodyDivs}{mmmmmm}{
  \luadirect{ekdosis.mkdivdepths(
    \luastringN{#1},
    \luastringN{#2},
    \luastringN{#3},
    \luastringN{#4},
    \luastringN{#5},
    \luastringN{#6}
    )
  }
}
```

Divisions specific to ekdosis. Define keys to be used by `\ekddiv`:—

```
\ekvdefinekeys{ekd@div}{
  code type = \def\type@value{#1},
  code n = \def\n@value{#1},
  code head = \def\head@value{#1},
  code barehead = \def\barehead@value{#1},
  store depth = \depth@value,
  code mark = \ekd@storemark{#1},
  choice toc = {book = \def\toc@value{book},
                part = \def\toc@value{part},
                chapter = \def\toc@value{chapter},
                section = \def\toc@value{section},
                subsection = \def\toc@value{subsection},
                subsubsection = \def\toc@value{subsubsection},
                paragraph = \def\toc@value{paragraph},
                subparagraph = \def\toc@value{subparagraph}},
  unknown-choice toc = \PackageError{ekdosis}{unknown toc=#1}{`toc'
    must be either `book', `part', `chapter', `section', `subsection',
    \MessageBreak `subsubsection', `paragraph' or `subparagraph'.},
  initial depth = 1
}
```

`\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 LaTeX formatting commands to go before and after the heading itself:—

```
\NewDocumentCommand{\FormatDiv}{m m m}{
  \luadirect{ekdosis.fmtdiv(\luastring{#1},
    \luastringN{#2},
    \luastringN{#3})}
}
```

`\ekd@getfmtdiv` gets the formatting commands that have been stored by `\FormatDiv`.

```
\NewDocumentCommand{\ekd@getfmtdiv}{m m}{%
  \luadirect{tex.sprint(ekdosis.getfmtdiv(\luastring0{#1},
    \luastringN{#2}))}%
}
```

`\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.

```
\NewDocumentCommand{\ekddiv}{m}{
  \begingroup
  \ekvset{ekd@div}{#1}%
  \ifdefined\head@value
    \bgroup
      \ekd@getfmtdiv{\depth@value}{b}%
      \head@value
      \ekd@getfmtdiv{\depth@value}{e}%
    \egroup
      \ifdefined\toc@value
        \ltx@ifpackageloaded{hyperref}{\phantomsection}{}%
          \ifdefined\barehead@value
            \addcontentsline{toc}{\toc@value}{\barehead@value}%
          \else
            \addcontentsline{toc}{\toc@value}{\head@value}%
          \fi
      \fi
  \fi
  \endgroup
}
```

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.

```
\newif\ifekd@test@vpnum
\newcounter{ekd@vpnum}
\globalcounter{ekd@vpnum}
\NewDocumentCommand{\test@vpnum}{}{%
  \ifekd@test@vpnum
    \edef\@tempa{\theekd@vpnum}%
    \stepcounter{ekd@vpnum}%
```

```
    \label{vpnum:\theekd@vpnum}%
    \ifnum
      \pdf@strcmp{\getpagerefnumber{vpnum:\@tempa}}%
        {\getpagerefnumber{vpnum:\theekd@vpnum}}
        = 0
      \else
        \resetvlinenumber
    \fi
  \else
  \label{vpnum:\theekd@vpnum}%
  \global\ekd@test@vpnumtrue
  \fi
}
```

\\+ \\+ 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. (Additionally, this command will be patched below to allow for maxlines definition.)

```
\ifboolexpr{bool {@pkg@poetry@verse} or bool {ekd@memoir@loaded}}
 {\newcommand{\@vsifplus}[1]{\@ifnextchar +{\@firstoftwo{#1}}}
   \renewcommand{\@vscentercr}{%
     \ifhmode \unskip\else \@nolnerr\fi
     \@vsifgt{\ifnum@brokenline\@vstypelinenum\fi\verselinebreak}{%
       \@vsifplus{\ifnum@brokenline\@vstypelinenum\fi\stepcounter{vslineno}%
         \par\@ifstar{\nobreak\@vsxcentercr}{%
           \@vsifbang{\@ifnextchar[ {\@vsicentercr}{}}{\@vsxcentercr}%
         }%
       }{%
         \@vstypelinenum
         \incr@vsline%
         \par\@ifstar{\nobreak\@vsxcentercr}{%
           \@vsifbang{\@ifnextchar[ {\@vsicentercr}{}}{\@vsxcentercr}%
         }%
       }%
     }%
   }
 }{}
```

Two small patches are applied to the verse package, then ekdverse is defined:—

```
\if@pkg@poetry@verse
```

This patch allows for maxlines to be used in verse environments:

```
\AddToHook{cmd/@vscentercr/before}{%
  \ifdefined\maxlines@value
    \stepcounter{ekd@lnperpage}%
    \ifnum\value{ekd@locallnperpage} = 1
      \ifnumcomp{\theekd@lnperpage}{=}{%
          \luadirect{tex.sprint(ekdosis.getlocalmaxlines())}}{%
          \setcounter{ekd@locallnperpage}{0}%
          \setcounter{ekd@lnperpage}{0}\pagebreak}{}%
      \else
        \ifnumcomp{\theekd@lnperpage}{=}{\maxlines@value}{%
            \setcounter{ekd@lnperpage}{0}\pagebreak}{}%
        \fi
```

```
      \fi}
\patchcmd{\start@vsline}{%
  \ifaltindent}{%
  \ifekd@pagevlineation\test@vpnum\fi
  \ifaltindent}{}{}
\ekvdefinekeys{ekd@verse}{
  dimen width = \vwidth@val,
  initial width = \linewidth,
  code type = \def\type@value{#1},
}
\ifekd@memoir@loaded
  \def\vlvnumfont{\textdir TLT\normalfont\footnotesize}
  \def\verselinenumfont#1{\def\vlvnumfont{#1}}
\else
  \verselinenumfont{\textdir TLT\normalfont\footnotesize}
\fi
\setcounter{poemline}{1}
\NewDocumentEnvironment{ekdverse}{!O{}}{%
  \ekvset{ekd@verse}{#1}%
  \if@continuous@vnum\setverselinenums{\thelinenumber}{0}\fi
  \nolinenumbers
  \let\linelabel\label
  \ifekd@memoir@loaded
    \refstepcounter{verse}%
  \else
    \stepcounter{verse@envctr}%
  \fi
  \addtocounter{poemline}{-1}\refstepcounter{poemline}%
  \setcounter{vslineno}{1}%
  \let\\=\@vscentercr
  \list{}{\itemsep \z@
    \itemindent  -\vindent%
    \listparindent\itemindent
    \parsep       \stanzaskip
    \setlength{\itemsep}{0pt}%
    \setlength{\topsep}{0pt}%
    \setlength{\partopsep}{0pt}%
    \ifdim\vwidth@val < \linewidth
      \rightmargin        \z@
      \setlength{\leftmargin}{\linewidth}%
      \addtolength{\leftmargin}{-\vwidth@val}%
      \addtolength{\leftmargin}{-0.5\leftmargin}%
    \else
      \rightmargin        \leftmargin
    \fi
    \addtolength{\leftmargin}{\vindent}%
  }%
\item[]\ifekd@pagevlineation\test@vpnum\fi%
}
{\endlist
  \if@continuous@vnum\resetlinenumber[\thepoemline]\fi}
```

Finally, this is the standard `verse` environment:—

```
\else
\newlength{\ekdverseindentlength}
\setlength{\ekdverseindentlength}{\parindent}
```

```
\NewDocumentEnvironment{ekdverse}{!O{\ekdverseindentlength}}{
  \begin{list}{}{%
      \setlength{\leftmargin}{#1}
      \setlength{\itemsep}{0pt}
      \setlength{\topsep}{0pt}
      \setlength{\partopsep}{0pt}
    }
  \item[]
  }{\end{list}}
\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.

```
\NewDocumentCommand{\resetvlinenumber}{O{1}}{%
  \if@pkg@poetry@verse
  \setverselinenums{#1}{0}%
  \fi
}
```

`ekdstanza` (*env.*) `ekdstanza` is needed when lines are grouped into stanzas, which can be further named by means of the `type` optional argument:—

```
\ekvdefinekeys{ekd@stanza}{
  code type = \def\type@value{#1}
}
\NewDocumentEnvironment{ekdstanza}{!O{}}{%
  \resetvlinenumber[\thepoemline]%
  \leavevmode\unskip
  \ekvset{ekd@stanza}{#1}%
  \ignorespaces
}{}
```

`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.

```
\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.

```
\newlength{\ekdspace}
\settowidth{\ekdspace}{ }
\def\ekdunspace{\hskip-\ekdspace}
```

Errors and Warnings

```
\def\ekd@wrong@ilabel{
  \PackageError{ekdosis}%
  {Unknown ``ilabel'' in \string\ilabel{}}%
  {Please check for an ``ilabel'' that exists.}
}
```

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.

```
\IfFileExists{\jobname-ekd.cfg}{\input{\jobname-ekd.cfg}}{}
```

20 Change History

21 Index

Numbers written in italic 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.

D

E

F

G

H

I

J

K

L

M

N

O

options:

P

Q

R

S

T