

V1.0 2024-10-06*

©2024 by Pablo González†

CTAN: https://www.ctan.org/pkg/enumext
 https://github.com/pablgonz/enumext

Abstract

This package provides enumerated list environments compatible with LTEX tagging PDF for creating "simple exercise sheets" along with "multiple choice questions", storing the "answers" to these in memory using multicol and scontents packages and the l3seq and l3prop modules.

C	ontents				_
1	Introduction	1	6	The storage system	11
	1.1 Description and usage	2		6.1 Keys for storage system	11
	1.2 The concept of left margin	3		6.1.1 Keys for label and ref	
	1.3 User interface	3		6.1.2 Keys for wrap and display	
	1.3.1 Internal counters	3		6.1.3 Keys for debug and checking	
	1.3.2 Public dimension	3		6.2 The command \anskey	
	1.3.3 Support for multicol	3			
	1.3.4 Support for minipage	4		6.2.1 Keys for \anskey	
	1.3.5 The \label and \ref system	4		6.3 The environment anskey*	
	1.3.6 Support for \footnote	4		6.3.1 Keys for anskey*	
2	The environments provided	5		6.4 The environment keyans	15
	2.1 The environment enumext	5		6.4.1 The \item* in keyans	15
	2.2 The environment enumext*	5		6.5 The environment keyanspic	16
	2.3 The command \item*	5		6.5.1 Keys for keyanspic	16
	2.3.1 Keys for \item*	6		6.5.2 The command \anspic	16
	2.4 The command \item in enumext* The command \setenumext	6		6.6 Printing stored content	
3	The command \setenumext The command \setenumextmeta	6		6.6.1 The command \getkeyans	
4 5	The keyval system	6 7		6.6.2 The command \foreachkeyans .	
J	5.1 Keys for label and ref	7		6.6.3 The command \printkeyans	
	5.2 Keys for spaces	8	7	Full examples	10
	5.2.1 Vertical spaces	8	8	Tagged PDF examples	
	5.2.2 Horizontal spaces	9	9	The way of non-enumerated lists	
	5.3 Keys for add code	9	,	References	
	5.4 Keys for start, series and resume			Change history	
	5.5 Keys for multicols			Index of Documentation	
	5.6 Keys for minipage				
	5.6.1 The command \miniright			Implementation	
	5.6.2 The key mini-right	11	14	Index of Implementation	141

Motivation and acknowledgments

Usually it is enough to use the classic enumerate environment to generate "simple exercise sheets" or "multiple choice questions", the basic idea behind enumext is to cover three points:

- 1. To have a simple interface to be able to write "lists of exercises" with "answers".
- 2. To have a simple interface for writing "multiple choice questions".
- 3. To have a simple interface for placing "columns" and "drawings" or "tables".

This package would not be possible without Phelype Oleinik who has collaborated and adapted a large part of the code and all Large Y team for their great work and to the different members of the TeX-SX community who have provided great answers and ideas. Here a note of the main ones:

- 1. Answer given by Alan Munn in \topsep, \itemsep, \partopsep, \parsep what do they each mean (and what about the bottom)?
- 2. Answer given by Enrico Gregorio in Understanding minipages aligning at top
- $_{\rm 3.}\,$ Answer given by Ulrich Diez in Different mechanics of hyperlink vs. hyperref
- 4. Answer given by Enrico Gregorio in Minipage and multicols, vertical alignment

 $^{^{*}}$ This file describes a documentation for v1.0, last revised 2024-10-06.

[†]E-mail: «pablgonz@educarchile.cl».

enumext v1.0 §.1 Introduction

License and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the LaTeX Project Public License (lppl), version 1.3 or later (https://www.latex-project.org/lppl.txt). The software has the status "maintained".

The enumext package loads and requires multicol[3] and scontents[4] packages, need to have a modern TEX distribution such as TEX Live or MiKTEX. It has been tested with the standard classes provided by ETEX: book, report, article and letter on 10pt, 11pt and 12pt.

1 Introduction

In the Large world world there are many useful packages and classes for creating "lists of exercises", "worksheets" or "multiple choice questions", classes like exam[1] and packages like xsim[2] do the job perfectly, but they don't always fit the basic day to day needs.

In my work (and in the work of many teachers) it is common to use "simple exercise sheets" also known as "informal lists of exercises", as an example:

- 1. Factor $x^2 2x + 1$
- 2. Factor 3x + 3y + 3z
- 3. True False
 - (a) $\alpha > \delta$
 - (b) LaTeX2e is cool?
- 4. Related to Linux

- (a) You use linux?
- (b) Usually uses the package manager?
- (c) Rate the following package and class
 - i. xsim-exam
 - ii. xsim
 - iii. exsheets

Sometimes we are also interested in showing the "answers" along with the questions:

- 1. Factor $x^2 2x + 1$
- $*|(x-1)^2$
- 2. Factor 3x + 3y + 3z
- $* \left| 3(x+y+z) \right|$
- 3. True False
 - (a) $\alpha > \delta$
 - * False
 - (b) LaTeX2e is cool?
 - * Very True!
- 4. Related to Linux

- (a) You use linux?
- * Yes
- (b) Usually uses the package manager?
 - * Yes, dnf
- (c) Rate the following package and class
 - i. xsim-exam
 - * doesn't exist for now :(
 - i. xsim
 - * very good
 - iii. exsheets
 - * obsolete

Or we are interested in referring to a specific question and its "answer", for example:

The answer to 3.(b) is "Very True!" and the answer to 4.(c).ii is "very good".

Or we are interested in printing all the "answers":

- 1. $(x-1)^2$
- 2. 3(x+y+z)
- 3. (a) False
 - (b) Very True!
- 4. (a) Yes

* (b) Yes, dnf

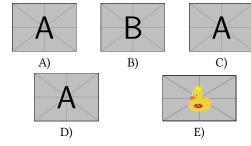
:

- - ii. very good
 - iii. obsolete
 - iii. obsolete

Another very common thing to use in my work is "multiple choice questions", for example:

- 1. First type of questions
 - A) value
- C) value
- B) correct
- D) value
- 2. Second type of questions
 - $I. 2\alpha + 2\delta = 90^{\circ}$
 - II. $\alpha = \delta$
 - III. $\angle EDF = 45^{\circ}$
 - A) I only
- D) I and III only
- B) II only
- E) I, II, and III
- C) I and II only
- * 3. Third type of questions
 - (1) $2\alpha + 2\delta = 90^{\circ}$
 - (2) $\angle EDF = 45^{\circ}$
 - (1) $2\alpha + 20 = 30$
 - A) value
- D) value E) value
- B) value
- C) value

4. Question with image and label below:



- 5. Question with image on left side:
 - A) value
 - B) value
 - C) value
 - D) correct
 - E) value



Where what we are interested in the $\langle label \rangle$ and a "short note" that we leave as an explanation, and then print them:

```
      1. B) x = 5
      * 4. E) A duck

      2. D)
      * 5. D) "other note"

      3. C) some note
      *
```

These "simple worksheets" or "multiple choice questions" appear to be easy to obtain using a combination of the enumerate, minipage and multicols environments, but like many things, what "looks simple" is not so simple.

The enumext package was created and designed to meet these small requirements in the creation of "simple worksheets" and "multiple choice questions".

1.1 Description and usage

The enumext package defines enumerated environments using the list environment provided by LTEX, but "does not redefine" any internal commands associated with it such as \list, \endlist or \item outside of the "scope" in which they are defined.

This package is NOT intend to replace the enumerate environment nor replace the powerful enumitem[6], the approach is intended to work without hindering either of them.

This package can be used with xelatex, lualatex, pdflatex and the classical latex»dvips»ps2pdf and is present in TeX Live and MiKTeX, use the package manager to install. For manual installation, download enumext.zip and unzip it, run lualatex enumext.dtx and move all files to appropriate locations, then run mktexlsr. To produce the documentation run lualatex enumext.dtx two times.

The package is loaded in the usual way:

```
\usepackage{enumext}
```

1.2 The concept of left margin

There is a direct relationship between the parameters \leftmargin, \itemindent, \labelwidth and \labelsep plus an "extra space" that makes it difficult to obtain the desired horizontal spaces in a list environment.

Usually we don't want the list to go beyond the left margin of the page, but since these four values are related, that causes a problem. The enumitem[6] package adds the \labelindent parameter to solve some of these problems. A simplified representation of this in the figure 1.



Figure 1: Representation of horizontal lengths in enumitem.

The enumext package does NOT provide a user interface to set the values for \leftmargin and \itemindent, instead it provides the keys list-offset and list-indent which internally set the values for \leftmargin and \itemindent. The concepts of \leftmargin and \itemindent are different in enumext. The figure 2 shows the visual representation of idea.



Figure 2: Representation of horizontal lengths concept in enumext.

In this way we reduce a *little* the amount of parameters we have to pass. With the default values of keys list-offset, list-indent, labelwidth and labelsep the lists will have the (usually) expected output for "simple worksheets". The figure 3 shows the visual representation.



Figure 3: Default horizontal lengths list-offset=0pt, list-indent=\labelwidth+\labelsep in enumext.

1.3 User interface

The user interface consists of two main list environments enumext (vertical) and enumext* (horizontal), the environment anskey* and the command \anskey to "store content" and the environments keyans, keyans* and keyanspic for multiple choice. It also provides the commands \getkeyans to print individual stored content, \printkeyans to print all stored content, \miniright for minipage and \setenumext to config all $[\langle key = val \rangle]$ options.

1.3.1 Internal counters

The package enumext uses internally the enumXi, enumXii, enumXii, enumXiv counters for the four nesting levels of the enumext environment, the enumXv counter for the keyans environment, the enumXvi counter for the keyanspic environment, the counter enumXviii for enumext* environment and the counter enumXviii for keyans* environment.

ilf any package defines these counters or they are user-defined in the document, the package will return a fatal error and abort the load.

1.3.2 Public dimension

The package enumext only provides a single public dimension \itemwidth and is intended for user convenience only and is not for internal use as such. The dimension \itemwidth is rigid length and contains the "width of the content" of each \item regardless of labelwidth and labelsep.

If any package defines \itemwidth or they are user-defined \itemwidth in the document, the package will overwrite it without warning.

1.3.3 Support for multicol

The package provides direct support for using the multicol[3] package. This allows to obtain directly a two-column output as shown in the figure 4.



Figure 4: Representation of the two column output for a nested level in enumext environment.

The "non starred" version of the multicols environment is always used together with the \raggedcolumns command and is controlled by columns and columns-sep keys. It can be used in all nesting levels of the environment enumext and the environment keyans and can together with the mini-env key. If you need to force a start a new column \columnbreak must be used (see §5.5).

The \columnseprule command is not available as a key and is set to "zero" for the inner levels and the keyans environment. If the value of this is set inside the document, it will affect "all environments" that use the columns key.

1.3.4 Support for minipage

The package provides direct support for minipage environment, this allows you to obtain an output like the one shown in figure 5.



Figure 5: Representation of the mini-env output for a nested level enumext environment.

The minipage environments on "left side" and "right side" is always used with "aligned on top" [t]. It can be used in all nesting levels of the environment enumext and the environment keyans and is controlled by mini-env and mini-sep keys. In order to switch from the "left" side minipage environment to the "right" side one must use the command \miniright (see §5.6).

1.3.5 The \label and \ref system

This package provides a user interface like the enumitem[6] package to customize the references which is activated by the ref key (§5.1), the standard ETEX \label and \ref commands work as usual. It also provides an "internal reference" system for the "stored content" by means of the key save-ref (§6.1.1) when the key save-ans (§6.1) is active.

1.3.6 Support for \footnote

This package provides an internal implementation for the \footnote command which is compatible with the hyperref package for the enumext* and keyans* environments, but will not produce the expected links, and if the mini-env key is used in enumext or keyans environments the output will look like the classic way they are displayed in the environment minipage.

The best way to solve this is to use Jean-François Burnol footnotehyper[9] package, it will support keeping the links if hyperref is loaded with the hyperfootnotes=true option (default) and will show the output numbered at the bottom of the page (as opposed to how it is displayed in the minipage environment). The way to load it is as follows:

```
\usepackage{footnotehyper}
\makesavenoteenv{enumext}
\makesavenoteenv{enumext*}
```

At the moment the footnotehyper package is not compatible with tagged PDF.

The environments provided

The package enumext provides two main list environments, the vertical environment enumext and the horizontal environment enumext*.

```
enumext*
```

```
\begin{enumext*} [\langle keyval \ list \rangle]
                     \item \langle item content \rangle
                                                                                                                                \item \(\(\)item \(\)content\\)
                      \item [\langle custom \rangle] \langle item content \rangle
                                                                                                                                \item [\langle custom \rangle] \langle item content \rangle
                                                                                                                                \verb|\item*| | \langle \mathit{symbol} \rangle | | \langle \mathit{offset} \rangle | | \langle \mathit{item content} \rangle
                      \left\langle item^* \left[ \left\langle symbol \right\rangle \right] \left[ \left\langle offset \right\rangle \right] \right\rangle
                  \end{enumext}
                                                                                                                             \end{enumext*}
```

The environment enumext 2.1

The enumext is an environment that works in the same way as the standard enumerate environment provided by ETEX, \item and \item[$\langle custom \rangle$] commands work in the usual way. The environment can be nested with at most "four levels" and the options can be configured globally using \setenumext command and locally using $[\langle key = val \rangle]$ in the environment.

Example with columns=2

1. This text is in the first level.

A. This text is in the fourth level.

- (a) This text is in the second level.
- X This text is in the first level.
- This text is in the third level.
- * 2. This text is in the first level.

The environment enumext*

The enumext* is a horizontal list environment similar to the enumerate* environment provided by the enumitem package or task environment provided by the task package, \item and \item[\langle custom \rangle] work as usual. The options can be configured globally using \setenumext command and locally using $[\langle key = val \rangle]$ in the environment.

Some considerations to take into account for this environment:

- The environment cannot be nested within itself or in the environment keyans*, but it can be nested within enumext and vice versa.
- Each "item content" in the environment is placed within a minipage environment whose width is stored in the dimension \itemwidth that NOT includes labelwith, labelsep, only the width of the content.
- You cannot have floating environments like figure or table but \footnote with hyperref support is supported if the footnotehyper package is loaded.
- · You cannot have any standard list environments like itemize, enumerate, description, quote, quotation, verse, center, flushleft, flushright, verbatim, tabbing, trivlist, list and all environments created with \newtheorem.

Example with columns=2

1. This text is in the first level.

2. This text is in the first level.

X This text is in the first level.

* 4. This text is in the first level.

The command \item*

```
\item* \item*
```

```
\forall item*[\langle symbol \rangle]
\time ' [\langle symbol \rangle] [\langle offset \rangle]
```

The $\idesign \ \$ and $\idesign \ \$ and $\idesign \ \$ works like the numbered $\idesign \ \$ works like the numbered $\idesign \ \$ a $\langle symbol \rangle$ to the "left" of the $\langle label \rangle$ separated from it by the $\langle offset \rangle$ set by the the second optional argument. The default values for $\langle symbol \rangle$ and $\langle offset \rangle$ are \textasteriskcentered '* and the value set by labelsep

The starred argument '*' cannot be separated by spaces 'u' from the command, i.e. \item* and the first optional argument does "NOT" support verbatim content. Can be configure with the keys item-sym* and item-pos* locally in the environment or globally using \setenumext command (§3).

■ The behavior of \item* in the enumext and enumext* environments is NOT the same as in the keyans and keyans* environments.

2.3.1 Keys for \item*

```
item-sym* = \{\langle symbol \rangle\}
```

default: \textasteriskcentered

Sets the *symbol* to be displayed in the "left" of the box containing the current $\langle label \rangle$ set by labelwidth key for \item* in enumext and enumext*. The symbol can be in text or math mode, for example item $sym*={\$\setminus star\$}.$

```
item-pos* = {\langle rigid \ length \rangle}
```

default: by levels

Sets the *offset* between the box containing the current $\langle label \rangle$ defined by labelwidth key and the $\langle symbol \rangle$ set by item-sym* key. The default values are set by labelsep key at each level. If positive values are passed it will offset to the left and if negative values are passed it will offset to the right.

The command \item in enumext*

The \item command for the enumext* environment provides an "first optional argument" \item(\langle columns \rangle) which "joins items" between columns. Let's consider the following examples adapted directly from the task package:

```
\begin{enumext*}[widest=10,columns=4]
  \item The first
  \item* The second
  \item The third
  \item The fourth
  \forall (3)* The fifth item is way too long for this and needs three columns
  \item The sixth
  \item The seventh
  \item(2)[X] The eighth item is way too long for this and needs two columns
    (\the\itemwidth)
 \item The ninth
 \item[Z] The tenth (\the\itemwidth)
\end{enumext*}
```

- 1. The first
- * 2. The second
- 3. The third
- 4. The fourth
- * 5. The fifth item is way too long for this and needs three columns
- 6. The sixth
- X The eighth item is way too long for this and needs 9. The ninth two columns (196.17749pt)
- The tenth (89.28171pt)

The command \setenumext

```
\setenumext \setenumext{\langle key = val \rangle}
                                                                                                                                         \star{keyans*} \{\langle keyans* \rangle\}
                          \strut = \sum \{\langle enumext, level \rangle \} \{\langle key = val \rangle \}
                                                                                                                                         \start \setenumext[\langle print, level \rangle] {\langle key = val \rangle}
                          \startion{1}{\text{setenumext}[\langle enumext^* \rangle] \{\langle key = val \rangle\}}
                                                                                                                                          \startion{1}{\text{setenumext}[\langle print, * \rangle] \{\langle key = val \rangle\}}
                          \star{\exists keyans} \ \{ \langle key = val \rangle \}
                                                                                                                                          \startion{1}{\text{setenumext}}[\langle print^* \rangle] \{\langle key = val \rangle\}
```

The command \setenumext sets the $\langle keys \rangle$ on a global basis for environments enumext, enumext*, keyans, keyans* and the \printkeyans command. It can be used both in the preamble and in the body of the document as many times as desired.

The \(\lambda \text{keys} \rangle \) set in the optional argument of environments and commands have the highest precedence, overriding both options passed by \setenumext. If the optional argument is not passed, the first level of the environment enumext will be taken by default.

🍼 The key save-ans that activate the *"storage system"* must NOT be passed through this command and must be passed directly in the optional argument of the "first level" of the environment in which they are executed.

The command \setenumextmeta

```
\setenumextmeta \setenumextmeta \{\langle key \ name \rangle\} \{\langle key \ one = val, \ key \ two = val, \ldots \rangle\}
                     \setenumextmeta*{\langle key name \rangle}{\langle key-one = val, key-two = val, ... \rangle}
                     \setenumextmeta [\langle enumext^* \rangle] \{\langle key \ name \rangle\} \{\langle key \ one = val, \ key \ two = val, \dots \rangle\}
```

The command \setenumextmeta adds a new "meta-key" for the environments enumext and enumext*, the {\langle key name \rangle} must be different from those defined by the package. If the optional argument is not passed, the new "meta-key" will be created for the "first level" of the environment enumext.

The starred argument '*' will create the new "meta-key" for the environment enumext* and for all levels of the environment enumext. For example: \setenumextmeta*{midsep}{topsep=3pt, partopsep=0pt} will create a new key midsep available for all levels of the enumext environment and the enumext* environment and we can use it like any other key so \begin{enumext} [midsep] and \begin{enumext*} [midsep] will be valid.

5 The keyval system

The $\langle key = val \rangle$ system used by the enumext package is implemented using lakeys so it must be taken into consideration that those keys marked as "value forbidden", that is $\langle key \rangle$ is different from $\langle key = \rangle$.

All \(\lambda keys\rangle\) described in this section are available for the enumext, enumext*, keyans and keyans* environments with the exception of the keys series, resume, resume* which are only available for the "first level" of the environments enumext and enumext*; and the keys mini-right, mini-right* which are only available for the enumext* and keyans* environments.

All $\langle keys \rangle$ related to vertical or horizontal spacing accept a "skip" or "dim" expression if passed between braces, i.e. you do not need to use \dimeval or \dimexpr to perform calculations.

It should be kept in mind that using any $\langle key \rangle$ that sets a *rubber lengths* or *rigid lengths* for vertical or horizontal space on a level will influence the vertical and horizontal space for *inners levels* and keyans, keyans* and keyanspic environments.

5.1 Keys for label and ref

mode-box (value forbidden)

default: not used

This is a "switch-key" that does not receive an argument and is "only" available for the "first level" of the enumext environment and the enumext* environment. When this is set the label, font, wrap-label and wrap-label* keys are executed within \makebox for the enumext and keyans environments.

- This key is intended for compatibility with tagged PDF and is forcibly "enabled" when \DocumentMetadata is present. If you want to get the same document output whether \DocumentMetadata is active or not, you must enable this key.
- In the enumext* and keyans* environments \makelabel are redefined using \makebox by default. If enumext or keyans is used in the enumext* environment the key must be activated manually.

```
label = \{ \langle \text{\ensuremath{}} alph^* | \text{\ensuremath{}} arabic^* | \text{\ensuremath{}} roman^* | \text{\ensuremath{}} Roman^* \rangle \}
```

default: by levels

Sets the $\langle label \rangle$ that will be printed at the *current level*. The default value for the first level of the environments enumext and enumext* are $\langle arabic^*, for second level$ are $\langle alph^* \rangle$, for third level are $\langle alph^* \rangle$. For keyans and keyans* environments the default value is $\langle alph^* \rangle$.

This key is intended to give the basic structure with which the $\langle label \rangle$ will be displayed, and the form in which it is used by standard "label and ref" and the "internal label and ref" system with the save-ref key. You cannot use commands with $\langle label \rangle$ as an argument, for example $\{ \text{nd} \mid \text{$

```
labelsep = \{ \langle rigid \ length \rangle \}
```

default: 0.3333em

Sets the *horizontal space* between the box containing the current $\langle label \rangle$ defined by label key and the text of an item on the first line. Internally sets the value of \labelsep for the current level.

```
labelwidth = \{\langle rigid \ length \rangle\}
```

default: by labe

Sets the width of the box containing the current $\langle label \rangle$ set by label key. Internally sets the value of \labelwidth for the current level. The default values are calculated by means of the width of a box by setting a value to the current counter using '0' for \arabic*, 'M' for \Alph*, 'm' for \alph*, 'VIII' for \Roman* and 'viii' for \roman*.

```
widest = \{ \langle integer \mid string \rangle \}
```

default: empty

Sets the labelwidth key pass the $\langle integer \rangle$ or converting the $\langle string \rangle$ of the form \Alph, \alph, \Roman or \roman to a value for the current counter defined by label key, then calculating the width by means of a box. For example widest={XXIII} or widest={23} are equivalent. This key is useful when the default values of the labelwidth key are smaller than those actually used.

```
font = \{ \langle font \ commands \rangle \}
```

default: empty

Sets the *font style* for the current $\langle label \rangle$ defined by label key. For example font={\bfseries\small}.

```
align = \{ \langle left \mid right \mid center \rangle \}
```

default: left

Sets the *aligned* of $\langle label \rangle$ defined by label key on the current level in the label box.

```
wrap-label = \{\langle code \{ \#1 \} \ more \ code \rangle \}
```

default: empty

Wraps the *current* $\langle label \rangle$ defined by label key referenced by $\{\#1\}$. The $\{\langle code \rangle\}$ must be passed between braces. This key does not modify the value set by the labelwidth key and is applied only on \item and \item*. When using it in the \setenumext command it is necessary to use the *double hash* ' $\{\#\#1\}$ '. For example wrap-label= $\{\fbox\{\#1\}\}\$ or you can create a command:

and then pass it through the key wrap-label={\labelbx{#1}} or wrap-label={\labelbx*{#1}}.

```
wrap-label* = \{\langle code \{\#1\} \mid more \ code \rangle\}
```

default: empty

The same as the wrap-label key but also applies on $\lceil \langle custom \rangle \rceil$.

```
©2024 by Pablo González L
```

```
ref = {\langle code \{ \lambda \} | \lambda \rangle |
```

default: empty

Modifies the way *cross references* are displayed. The label key sets the default form of the *cross references*, by using this key you can define a different format, for example: $ref=\ensuremath{\mathsf{ref}}\xspace \ensuremath{\mathsf{nef}}\xspace \ensuremath{\mathsf{ref}}\xspace \ensuremath{\mathsf{nef}}\xspace \ensuremath{\mathsf{nef}}\xspa$

Internally it renews the command associated with each counter when it is executed, i.e., in the environment enumext the command \theenumXi is modified when the key is executed at the first level, \theenumXii when it is executed at the second level and \theenumXiii together with \theenumXiv when it is executed at the third and fourth levels.

This must be kept in mind, since the values set by the label and ref keys are not cumulative by levels, so if you have used the ref key in the first level and then want to associate the counter with label or ref in the second level you must use the direct commands, i.e. \arabic{eunumXi} to indicate the count of the first level instead of using \theenumXi.

5.2 Keys for spaces

 $show-length = \{ \langle true \mid false \rangle \}$

default: false

Displays on the terminal the values for *all list parameters* at the current level. For *vertical spaces* show the values of \topsep, \itemsep, \parsep and \partopsep. For *horizontal spaces* show the values of \labelwidth, \labelsep, \itemindent, \listparindent and \leftmargin.

5.2.1 Vertical spaces

```
\texttt{topsep} = \{ \langle \mathit{rubber} \ \mathit{length} \mid \mathit{rigid} \ \mathit{length} \rangle \}
```

default: by levels

Set the *vertical space* added to both the top and bottom of the list. Internally sets the value of \topsep for the current level. The default value for the first level of the environments enumext and enumext* are 8.0pt plus 2.0pt minus 4.0pt, for second level are 4.0pt plus 2.0pt minus 1.0pt, for third and fourth level are 2.0pt plus 1.0pt minus 1.0pt. For keyans and keyans* environments the default value is 4.0pt plus 2.0pt minus 1.0pt.

```
parsep = \{ \langle rubber \ length \mid rigid \ length \rangle \}
```

default: by levels

Set the *vertical space* between paragraphs within an item. Internally sets the value of \parsep for the current level. The default value for the first level of the environments enumext and enumext* are 4.0pt plus 2.0pt minus 1.0pt, for second level are 2.0pt plus 1.0pt minus 1.0pt, for third and fourth level are 0pt. For keyans and keyans* environments the default value is 2.0pt plus 1.0pt minus 1.0pt.

In the enumext* and keyans* environments this value is passed to \parskip within the minipage environment where "item content" is placed.

```
partopsep = \{ \langle rubber \ length \mid rigid \ length \rangle \}
```

default: by levels

Set the *vertical space* added, beyond topsep, to the "top" and "bottom" of the entire environment if the environment instance is preceded by a "blank line" or \par command. Internally sets the value of \partopsep for the current level. The default values for first and second level in environment enumext are 2.0pt plus 1.0pt minus 1.0pt, for third and fourth level are 1.0pt minus 1.0pt. For the keyans environment the default value is 2.0pt plus 1.0pt minus 1.0pt, and for the keyans* and enumext* environments it is available but *without* effect.

The value of this parameter also affects the *inner levels* and the environments keyans, keyanspic and keyans*. Caution should be taken with "blank lines" or \par command "before" each environment or nested level when formatting the source code of document. Tex will enter (vertical mode) and apply this value to the "top" and "bottom" the environment or nested level.

```
itemsep = \{ \langle rubber \ length \ | \ rigid \ length \rangle \}
```

default: by levels

Set the *vertical space* between items, beyond the parsep. Internally sets the value of \itemsep for the current level. The default value for the first level of the environments enumext and enumext* are 4.0pt plus 2.0pt minus 1.0pt, for the rest of the levels are 2.0pt plus 1.0pt minus 1.0pt. For keyans and keyans* environments the default value is 4.0pt plus 2.0pt minus 1.0pt.

🍼 In the enumext* and keyans* environments this value corresponds to the separation between rows.

noitemsep \(\value forbidden \)

default: not used

This is a "meta-key" that does not receive an argument. Set itemsep and parsep equal to opt the entire level of environment.

nosep (value forbidden)

default: not used

This is a "meta-key" that does not receive an argument. Sets all keys for vertical spacing equal to opt the entire level of environment.

base-fix \(\text{value forbidden} \)

default: not used

This is a "switch-key" that does not receive an argument available only for the "first level" of environment enumext. Fix the baseline when an environment enumext is nested in enumext* and there is no material between the \item and the start of the environment for example \item \begin{enumext} enumext} within the environment enumext*. Internally sets the keys topsep, above and above* at Opt.

The following \(\lambda \text{keys} \rangle \) should be used with "caution", they are intended to be used at the "top" and "bottom" of the environment when the columns or mini-env keys do not provide adequate vertical spaces. The values passed can be rubber or rigid lengths, the way they are applied is the way you differ, using the star '*' \(\lambda \text{keys} \rangle \) applies \(\nabla \text{space} \text{* so that } \text{\text{MFX}} \text{ does not discard} \text{ this space at page break.} \)

```
above = \{\langle rubber\ length \mid rigid\ length \rangle\}
```

default: not used

Set the *extra vertical space* added, beyond topsep, to the top of the entire level of environment. This key is intended to give a *"fine adjustment"* of the vertical space *"above"* the environment without hindering the value of the topsep key. The space is added with \vspace so is *"discardable"*.

```
above* = {\langle rubber length | rigid length \rangle}
```

default: not used

Set the *extra vertical space* added, beyond topsep, to the top of the entire level of environment. This key is intended to give a *"fine adjustment"* of the vertical space *"above"* the environment without hindering the value of the topsep key. The space is added with \vspace* so is *"not discardable"*.

```
below = \{\langle rubber\ length \mid rigid\ length \rangle\}
```

efault: not used

Set the *extra vertical space* space added, beyond topsep, to the bottom of the entire level of environment. This key is intended to give a *"fine adjustment"* of the vertical space on the *"below"* the environment without hindering the value of the topsep key. The space is added with \vspace so is *"discardable"*.

```
below* = \{\langle rubber\ length \mid rigid\ length \rangle\}
```

default: not used

Set the *extra vertical space* space added, beyond topsep, to the bottom of the entire level of environment. This key is intended to give a *"fine adjustment"* of the vertical space on the *"below"* the environment without hindering the value of the topsep key. The space is added with \vspace* so is *"not discardable"*.

5.2.2 Horizontal spaces

 $list-offset = \{ \langle rigid \ length \rangle \}$

default: Opt

Sets the *horizontal translation* of the entire environment level from the left edge of the box defined by the labelwidth key. Internally sets the values of \leftmargin and \itemindent for the current level.

 $list-indent = \{ \langle rigid \ length \rangle \}$

default: labelwidth + labelsep

Sets the *indentation* of the whole environment under the box defined by labelwidth and labelsep keys. Internally sets the value of \leftmargin and \itemindent for the current level. If list-indent=0pt is set in the environments enumext and keyans the $\langle label \rangle$ will be part of the text, separated by the value of the labelsep key and the *first word*, in simple terms it will look like a "common paragraph".

The enumext* and kenyans* environments are implemented using \makebox and minipage which causes "list indent" to always be equal to the value passed to labewdith plus labelsep. Passing a value to this key is equivalent to setting the value for the list-offset key.

```
itemindent = \{ \langle \mathit{rigid} \; \mathit{length} \rangle \}
```

default: 0p

Sets the extra horizontal indentation, beyond labelsep, of the "first line" off each \item that is not followed by a "blank line" or the \par command. This value must be greater than or equal to opt and is applied internally using \hspace without modifying the value of \itemindent.

This key is intended for the enumext* and keyans* environments where, by their implementation, it is not possible to adjust labelwidth and list-indent without modifying the output. If you use enumext or keyans and want to get around the blank line limitation or the \par command followed by \item you can modify labelwidth and label-indent and get the same effect.

```
rightmargin = \{\langle rigid \ length \rangle\}
```

default: 0pt

Set the *horizontal space* between the right margin of the environment and the right margin of the enclosing environment, the value it takes must be greater than or equal to <code>Opt</code>. Internally sets the value of <code>\rightmargin</code> for the current level.

```
listparindent = \{\langle rigid\ length\rangle\}
```

default: 0pt

Sets the *horizontal space* indentation, beyond list-indent, for second and subsequent paragraphs within a list item. Internally sets the value of \listparindent for the current level.

In the enumext* and keyans* environments this value is passed to \parindent within the minipage environment where "item content" is placed.

5.3 Keys for add code

The following $\langle keys \rangle$ should be used with "caution", they are intended to inject $\{\langle code \rangle\}$ into different parts of the defined environments. We must keep in mind that the defined environments are based on the list base environment provided by ETEX which is defined (simplified) as plain form $\{ arg\ one \} \} \{\langle arg\ two \rangle \}$. Using the before* key does not allow access to the list parameters defined by $[\langle key=val \rangle]$.

```
before = \{\langle code \rangle\}
```

default: not used

Execute $\{\langle code \rangle\}$ "before" the environment starts. The $\{\langle code \rangle\}$ must be passed between braces, is executed "after" performing all calculations related to the *list parameters* in the environment and the parameters sets by $[\langle key = val \rangle]$ that is, in the second argument of the list after setting all the parameters \begin{list}{\langle arg one \rangle}{\langle arg two \rangle}{\langle code \rangle}}.

before* = $\{\langle code \rangle\}$ default: not used

Execute $\{\langle code \rangle\}$ "before" the environment starts. The $\{\langle code \rangle\}$ must be passed between braces, is executed "before" performing all calculations related to the *list parameters* and $[\langle key = val \rangle]$ sets in the environment that is, before the arguments defining the environment are executed: $\{\langle code \rangle\}\setminus\{arg\ one \}\}\{\langle arg\ one \rangle\}\{\langle arg\ one \rangle\}$.

 $first = \{\langle code \rangle\}$ default: not used

Executes $\{\langle code \rangle\}$ when "starting" the environment. The $\{\langle code \rangle\}$ must be passed between braces, is executed right "after" all list parameters are done, after the second argument of list, just before the first occurrence of \item: \begin{list}{\langle} \arg one \rangle \{\langle} \code \rangle \\ \item.

- Keep in mind that the code set in this key will affect the entire "body" of the environment and therefore the inner levels of the list and the keyans environment. It is recommended to set this key per level.
- In the enumext* and keyans* environments this key is executed after the listparindent, parsep and itemindent keys within the minipage environment in which the "item content" is placed.

 $\mathsf{after} = \{ \langle \mathit{code} \rangle \}$

Execute $\{\langle code \rangle\}$ "after" finishing the environment. The $\{\langle code \rangle\}$ must be passed between braces.

5.4 Keys for start, series and resume

$start = \{ \langle integer \mid integer \ expression \rangle \}$

default: 1

 $start* = \{ \langle integer \mid string \rangle \}$

default: not use

Sets the *start value* of the numbering on the current level. Internally $\langle string \rangle$ is converted and passed as value to the counter defined by label key on the current level, i.e. it is equivalent to enter start=5, start=E or start=v.

The following $\langle keys \rangle$ are "only" available for the enumext* environment and the "first level" of the enumext environment and are ignored if set when nested within each other.

series = $\{\langle series \ name \rangle\}$ default: not used

Stores the *keys* of the *optional argument* of the "first level" of the environment in which it is executed in $\{\langle series\ name \rangle\}$ which is used as an argument in the key resume. The $\langle keys \rangle$ stored in $\{\langle series\ name \rangle\}$ are not cumulative and are overwritten if the same $\{\langle series\ name \rangle\}$ is used again.

resume = $\{\langle series \ name \rangle\}$

default: not used

Sets the *start value* and *options* for the "*first level*" continuing the numbering of the environment in which the $series=\{\langle series\ name\rangle\}$ key was executed. If passed *without value* this will only set *start value* continue the numbering from the last environment in which $series=\{\langle series\ name\rangle\}$ or $resume=\{\langle series\ name\rangle\}$ is not present and if the save-ans key is active it will continue the numbering from the last environment in which it was executed. The *start value* can be overwritten using start or $start^*$ keys.

resume* \(\nabla value forbidden \rangle

default: not used

Sets the *start value* and *options* for the *"first level"* continuing the numbering of the environment in which the $series=\{\langle series\ name \rangle\}$ or $resume=\{\langle series\ name \rangle\}$ keys are NOT present, if the save-ans key is active it will continue the numbering from the last environment in which it was executed. The *start value* can be overwritten using start or $start^*$ keys.

For security reasons the series key will never save in {\series name\} the keys series, resume, resume*, save-ans, save-key, start* and start. When using the key resume={\series name\} it will have hierarchy in the \series keys\ that are saved in {\series name\}, in order to establish the value of a \series name\ along already saved in {\series name\} it must be placed to the "right" of resume={\series name\}, the same thing happens with the resume* key, the exception is the save-ans key that must be placed on the "left" if you want to start the numbering with its value. The resume key passed "without value" must be exactly "without value", i.e. resume= cannot be used and if executed before resume* it will affect the start value.

5.5 Keys for multicols

 $columns = \{\langle integer \rangle\}$ default: 1

Set the *number of columns* to be used by the multicols environment within the environment. The value must be a positive integer less than or equal to 10.

```
columns-sep = \{ \langle rigid \ length \rangle \}
```

default: by level

Set the *space between* columns used by the multicols environment within the environment. Internally sets the value of \columnsep, by default its value is equal to the sum of the values set in the keys labelwidth and labelsep of the current level.

- **③** The \footnote{ $\langle text \rangle$ } command in the nested levels of multicols will not work as expected, prefer the use of \footnotemark[$\langle number \rangle$] inside the environment and \footnotetext[$\langle number \rangle$] { $\langle text \rangle$ } outside the environment or via the after key.
- **③** By default the hyperref package does not provide support for \footnotemark[$\langle number \rangle$] and \footnotetext[$\langle number \rangle$] {\langle text\rangle}, but when tagged PDF is enabled and the package is loaded the \footnotemark[$\langle number \rangle$] and \footnotetext[$\langle number \rangle$] {\langle text\rangle} commands create links correctly.

©2024 by Pablo González L

5.6 Keys for minipage

```
mini-env = \{\langle rigid\ length \rangle\}
```

default: not used

Sets the width of the minipage environment on the "right side". This value added to the value set by the mini-sep key to determines the width of the minipage environment on the "left side", taking \linewidth as the maximum reference value.

```
mini-sep = \{\langle rigid \ length \rangle\}
```

default: 0.3333em

Sets the space between the minipage environment on the "left side" and the minipage environment on the *"right side"*. This separation is applied together with \hfill.

5.6.1 The command \miniright

```
\begin{enumext} [mini-env=\langle rigid length \rangle ] \langle item's before \rangle item \miniright*\langle content \rangle \text{end} \end{enumext}
```

The \miniright command close the minipage environment on the "left side" and opens the minipage environment on the "right side" by starting it with the \centering command. It must be placed "after" the last \item of the current environment and "before" starting the material to be placed on the "right side".

The starred argument '*' inhibits the use of \centering command i.e. the usual LTEX justification is maintained in the minipage on the "right side".

- of The \footnote $\{\langle text \rangle\}$ command in minipage environment will work as usual. If you prefer the footnotes to be numbered (not lowercase) and outside the environment, use $\footnotemark[\langle number \rangle]$ inside the environment and $footnotetext[\langle number \rangle] \{\langle text \rangle\}$ outside the environment or via the after key (see §1.3.6 for full support).
- ◆ By default the hyperref package does not provide support for \footnotemark[⟨number⟩] and \footnotetext[⟨number] $\{\langle text \rangle\}$, but when tagged PDF is enabled and the package is loaded the \footnotemark[$\langle number \rangle$] and \footnotetext[$\langle number \rangle$] { $\langle text \rangle$ } commands create links correctly.

5.6.2 The key mini-right

In the horizontal list environments enumext* and keyans* it is not possible to use the \miniright command and the mini-right key must be used instead.

```
mini-right = \{\langle content \rangle\}
```

default: not used

Set the *content* for the drawing or tabular to be placed in the minipage environment on the "right side" by starting it with \centering. The $\{\langle content \rangle\}$ must be passed between braces.

```
mini-right* = \{\langle content \rangle\}
```

default: not used

Same as above, but without starting with \centering.

The storage system

The entire mechanism for "storing content" it is activated according to save-ans key on the "first level" of enumext or enumext* environments and it is ignored if they are established when they are nested inside each other. Only when this $\langle key \rangle$ is "active" the \anskey command and the environments anskey*, keyans, keyans* and keyanspic are available.

```
\begin{enumext} [save-ans={\langle store name \rangle}]
                                                             \begin{enumext} [save-ans={\langle store name \rangle}]
  \item Text \anskey{answer}
                                                               \item Text \anskey{answer}
  \item Text
                                                               \item Text
    \begin{keyans}
                                                                 \begin{keyanspic}
    \end{keyans}
                                                                 \end{keyanspic}
\end{enumext}
                                                             \end{enumext}
```

By executing the key save-ans= $\{\langle store\ name \rangle\}$ the entire "structure" of the environment (excluding the first level) including the optional argument passed to the inner levels or the environment nested in it, along with the $\langle content \rangle$ passed to \anskey or anskey*, the current $\langle labels \rangle$ for \item* and \anspic* in the environments keyans, keyans* and keyanspic will be "stored" in a sequence $\{\langle store\ name \rangle\}$ and at the same time will be "stored" (without the "structure" or optional argument) in a prop list $\{\langle store\ name \rangle\}$.

For security reasons the optional argument of the inner levels or the nested environment are filtered by excluding all (keys) related to the "storage system" (§6.1) along with the keys mini-env, mini-sep, mini-right, miniright*, series, resume and resume* when storing in sequence {\store name\}} set by save-ans key.

Keys for storage system

The only \(\lambda \text{eys} \rangle \) available for all levels of the enumext environment and the enumext* environment are nostore and save-key, the rest of the $\langle keys \rangle$ described in this section must be passed directly in the *optional* argument of the "first level" of the environment in which the key save-ans is executed. The key save-ans should NOT be passed with the command \setenumext.

```
save-ans = \{ \langle store \ name \rangle \}
```

default: not set

Sets the *name* of the *sequence* and *prop list* in which the $\{\langle contents \rangle\}$ will be "stored" by \anskey and anskey* in enumext and enumext* environments and the current $\langle labels \rangle$ for \item* and \anspic* in the environments keyans, keyans* and keyanspic. If the *sequence* or *prop list* $\{\langle store\ name \rangle\}$ does not exist, it will be created globally and will not be *overwritten* if the key is used again.

```
save-key = \{\langle key \, list \rangle\}
```

default: not set

This key *overrides* the default "*stored keys*" of the *optional argument* of the inner levels or nested environment that will be passed to the *sequence*. The $\langle key \ list \rangle$ passed to this key ignores any $\langle keys \rangle$ in the "*stored structure*" and must be passed between braces. For example, if we execute at a second level:

```
\begin{enumext}[save-ans={\store name\}]
\item Text \anskey{answer}
\item Text
\begin{enumext}[nosep, columns=2, save-key={columns=3}]
...
\end{enumext}
\end{enumext}
```

The "stored keys" by default in the sequence $\{\langle store\ name \rangle\}$ would be nosep, columns=2, but using the key save-key= $\{columns=3\}$ will overwrite and the "stored key" in the sequence $\{\langle store\ name \rangle\}$ are only columns=3 ignoring all the others.

 $save-sep = \{ \langle text \ symbol \rangle \}$

default: {, }

Sets the *text symbol* that will separate the current $\langle label \rangle$ to the *optional argument* passed to the \item* and \anspic* in the environments keyans, keyans* and keyanspic and storing them in the *sequence* and *prop list* $\{\langle store\ name \rangle\}$ set by save-ans key. The $\{\langle text\ symbol \rangle\}$ must always be passed between braces, whitespace ' \sqcup ' is preserved within the braces and only affects the "stored content" and not what is displayed when using the show-ans or show-pos keys.

6.1.1 Keys for label and ref

 $save-ref = \{ \langle \mathit{true} \mid \mathit{false} \rangle \}$

default: false

Activates the "internal label and ref" mechanism for referencing "stored content" in prop list $\{\langle store\ name \rangle\}$ set by save-ans key. To reference the location of the "stored content" within the environment you must use $\texttt{ref}\{\langle store\ name:position \rangle\}$, where $\langle position \rangle$ corresponds to the position occupied by the "stored content" in the prop list $\{\langle store\ name \rangle\}$ returned by the show-pos key. For example $\texttt{ref}\{\texttt{test:4}\}$ will return 3. (b) which corresponds to the location of the "stored content" at position 4 in prop list test within the environment in which the key save-ans=test was set.

 $mark-ref = \{\langle symbol \rangle\}$

default: *

Sets the *symbol* that will be displayed by the \printkeyans command only if the hyperref package is detected and the save-ref key are active. This "*symbol*" is used as a "*link*" between the environment in which the save-ans key was used and the place where the command is executed.

6.1.2 Keys for wrap and display

```
wrap-ans = \{ \langle code \{ #1 \} \ more \ code \rangle \}
```

default: \fbox+\parbox{#1}

Wraps the *argument* passed to the \anskey and the *body* in anskey* environment referenced by $\{\#1\}$ when using the show-ans or show-pos keys. The $\{\langle code \rangle\}$ must be passed between braces and only affects the *argument* or *body* and NOT the "stored content" in the sequence and prop list $\{\langle store\ name \rangle\}$ set by save-ans key. If this key is passed using \setenumext it is necessary to use double ' $\{\#1\}$ '.

```
wrap-opt = \{ \langle code \{ #1 \} \ more \ code \rangle \}
```

default: [{#1}]

Wraps the *optional argument* passed to the \item* and \anspic* referenced by $\{\#1\}$ in the keyans, keyans* and keyanspic environments when using the show-ans or show-pos keys. The $\{\langle code \rangle\}$ must be passed between braces and only affects the current *optional argument* and NOT the "stored content" in the sequence and *prop list* $\{\langle store\ name \rangle\}$ set by save-ans key. If this key is passed using \setenumext it is necessary to use double ' $\{\#1\}$ '.

 $\mathsf{show-ans} = \{ \langle \mathit{true} \mid \mathit{false} \rangle \}$

default: false

Displays the *argument* passed to the \anskey, the *body* for anskey* environment, the $\langle label \rangle$ for \item* and \anspic* at the place where it is executed. If the *optional argument* is present in \item* or \anspic* it will be shown using wrap-opt key.

 $mark-ans = \{\langle symbol \rangle\}$

default: \textasteriskcentered

Sets the *symbol* to be displayed in the left margin for \anskey, anskey*, \item* and \anspic* in the place where they are executed when using the key show-ans.

 $mark-pos = \{\langle left \mid right \rangle\}$

default: left

Sets the *aligned* of the symbol defined by mark-ans key. The "symbol" is aligned in a box with the same dimensions of the label box defined by labelwidth key on the current level and separated by the value of the labelsep key.

6.1.3 Keys for debug and checking

 $show-pos = \{\langle true \mid false \rangle\}$

default: false

Displays the position occupied by the "stored content" by \anskey, anskey*, \item* and \anspic* in the prop list {\store name\} set by save-ans key. This position is used by the \getkeyans command and by the \ref command if the save-ref key is active.

check-ans = $\{\langle true \mid false \rangle\}$

Enables the checking answer mechanism displaying an appropriate message on the terminal. This key works under the logic that each \item or \item* that does not open an inner level or nested environment contains "only one answer" or "only one execution" of the \anskey or anskey*. It is intended to be used in conjunction with the no-store key.

no-store

default: not used

This is a *meta-key* that does not receive an argument and disables the structure stored in the *sequence* { (store name) set by save-ans key at the entire level or a nested environment in which it runs. This key is intended for use in internal levels or nested enumext or enumext* environments in which you want to use enumext or enumext* but "without" using the \anskey, "without" use anskey*, "without" interfering with the check-ans key and "without" storing an unwanted structure in the sequence $\{\langle store\ name \rangle\}$.

The command \anskey

\anskey \anskey [$\langle keys \rangle$] { $\langle content \rangle$ }

The command \anskey takes a mandatory non empty argument $\{\langle content \rangle\}$ and "stores" it in the sequence and prop list {\store name\} set by save-ans key. By design the command cannot be nested or passed verbatim material in the argument and it is assumed that each numbered \item or \item* within the environment in which it is active it has a "single execution" of \anskey unless \item or \item* open a nested level or use the no-store key.

If save-ref key are active and the <a href="https://hyperlink.gov/hyperl be used, otherwise the usual "label and ref" system provided by LATEX will be used.

The \anskey command is available for all levels of the enumext environment and the enumext* environment, but is disabled for the keyans, keyans* and keyanspic environments.

6.2.1 Keys for \anskey

By default the $\{\langle content \rangle\}$ passed to \anskey when "storing" in the sequence $\{\langle store\ name \rangle\}$ has the form \item $\langle content \rangle$, the following $\langle keys \rangle$ allow modifying the way in which it is "stored" in the sequence.

break-col (value forbidden)

default: not used

Stores $\{\langle content \rangle\}$ in the sequence $\{\langle store\ name \rangle\}$ of the form $\langle columnbreak \rangle$ item $\langle content \rangle$.

 $item-join = \{\langle columns \rangle\}$

default: not set

Set the *number of columns* to be used for $\forall item(\langle columns \rangle)$ and stores $\{\langle content \rangle\}$ in the *sequence* $\{\langle store \rangle\}$ name) of the form \item($\langle columns \rangle$) $\langle content \rangle$.

item-star \langle value forbidden \rangle

default: not used

Stores $\{\langle content \rangle\}$ in the sequence $\{\langle store\ name \rangle\}$ of the form $\backslash item^* \langle content \rangle$.

 $item-sym* = \{\langle symbol \rangle\}$

default: not set

Sets the symbol for $\forall tem^*$ when using the key $\exists tem^*$ and stores $\{\langle content \rangle\}$ in the sequence $\{\langle store, tem^* \rangle\}$ name) of the form \item*[$\langle symbol \rangle$] $\langle content \rangle$. The symbol can be in text or math mode, for example item-sym*= $\{\$\ast\$\}\ stores \item*[\$\ast\$] \ \langle content \rangle$.

 $item-pos* = \{\langle rigid\ length \rangle\}$

default: not set

Sets the *offset* for \forall when using the keys item-star and item-sym* and stores $\{\langle content \rangle\}$ in the *sequence* $\{\langle store\ name \rangle\}$ of the form $\idesign [\langle symbol \rangle] [\langle offset \rangle] \langle content \rangle$.

Example

```
\begin{enumext}[save-ans=test,show-ans=true]
  \item* Text containing our instructions or questions. \anskey\{\langle first\ answer \rangle\}
  \item Text containing our instructions or questions.
    \begin{enumext}
      \item Question.\anskey{\langle second \ answer \rangle}
    \end{enumext}
  \item Text containing our instructions or questions. \angle answer \
  \item Text containing our instructions or questions. \angle answer \
\end{enumext}
```

- * 1. Text containing our instructions or questions.
 - first answer
 - - (a) Question.
 - 2. Text containing our instructions or questions.
 - * second answer ©2024 by Pablo González L

- 3. Text containing our instructions or questions.
- third answer
- 4. Text containing our instructions or questions.
- fourth answer

6.3 The environment anskey*

 $anskey^* \setminus begin\{anskey^*\}[\langle key = val \rangle] \langle body content \rangle \setminus end\{anskey^*\}$

The environment anskey* takes a mandatory $\{\langle body\ content \rangle\}$ and "stores" it in the sequence and prop list $\{\langle store\ name \rangle\}$ set by save-ans key. If save-ref key are active and the hyperref[8] package is detected, hyperlink and hypertarget will be used, otherwise the usual "label and ref" system provided by ETEX will be used.

By design the environment cannot be nested but full supports "verbatim material" in the body and it is assumed that each numbered\item or \item* within the environment in which it is active it has a "single execution" unless \item or \item* open a nested level or use the no-store key.

The anskey* environment is implemented using the scontents package, for the correct operation \begin{anskey*} and \end{anskey*} must be in different lines, all $\langle keys \rangle$ must be passed separated by commas and "without separation" of the start of the environment. Comments "%" or "any character" after \begin{anskey*} or $[\langle key=val \rangle]$ on the same line are NOT supported, the package scontents will return an "error" message if this happens. In a similar way comments "%" or "any character" after \end{anskey*} on the same line the package scontents will return a "warning" message.

6.3.1 Keys for anskey*

The anskey* environment uses the same $\langle keys \rangle$ as the \anskey command next to the keys inherited from package scontents. The environment is available for all levels of the enumext environment and the enumext* environment, but it is disabled for the keyans, keyans* and keyanspic environments.

```
write-env = \{\langle file.ext \rangle\} default: not used
```

Sets the name of the $\langle external\ file \rangle$ in which the $\langle contents \rangle$ of the environment will be written. The $\langle file.ext \rangle$ will be created in the working directory, relative or absolute paths are not supported. If $\langle file.ext \rangle$ does not exist, it will be created or overwritten if the overwrite key is used.

```
overwrite = \{\langle true \mid false \rangle\} default: false

Sets whether the \langle file.ext \rangle generated by write-env from the anskey* environment will be rewritten.

force-eol = \{\langle true \mid false \rangle\} default: false
```

Sets if the *end of line* for the \(\stored content\) is hidden or not. This key is necessary only if the last line is the closing of some environment defined by the \(\frac{fancyvrb}{fancyvrb}\) package as \\end{\(\text{Verbatim}\)}\) or another environment that does not support a comments "%" after closing \\end{\(\text{Verbatim}\)}\%.

For security reasons the keys store-env, print-env and write-out they have been left disabled. It is recommended that you review the scontents[4] documentation to understand how the keys described here work.

Example

```
\begin{enumext}[save-ans=test,show-pos=true,start=5]
  \item* Text containing our instructions or questions.
    \begin{anskey*}[item-star]
      ⟨first answer⟩
    \end{anskey*}
  \item Text containing our instructions or questions.
    \begin{enumext}
      \item Question.
        \begin{anskey*}
          ⟨second answer⟩
        \end{anskey*}
    \end{enumext}
  \item Text containing our instructions or questions.
    \begin{anskev*}
      (third answer)
    \end{anskey*}
  \item Text containing our instructions or questions.
    \begin{anskey*}
      (fourth answer)
    \end{anskey*}
\ensuremath{\mbox{\mbox{enumext}}}
```

```
* 5. Text containing our instructions or questions.

[5] First answer with verbatim

6. Text containing our instructions or questions.

(a) Question.

[6] second answer

7. Text containing our instructions or questions.

[7] third answer

8. Text containing our instructions or questions.

[8] fourth answer
```

©2024 by Pablo González L 14/156

6.4 The environments keyans and keyans*

```
keyans \begin{keyans}[\langle key=val \rangle] \item \item[\langle custom \rangle] \item* \item*[\langle content \rangle] \end{keyans} keyans* \begin{keyans*}[\langle key=val \rangle] \item \item[\langle custom \rangle] \item* \item*[\langle content \rangle] \end{keyans*}
```

The keyans and keyans* environments are "enumerated list" environments designed for "multiple choice" questions activated by the save-ans key. This environments can NOT be nested and must always be at the "first level" of the enumext environment, the commands \item[$\langle custom \rangle$] work in the usual and the command \item($\langle columns \rangle$) is available for the keyans* environment.

```
\begin{enumext}[save-ans=test]
                                                                                           \begin{enumext} [save-ans=test]
   \item (item content)
                                                                                              \item \(\(\)item \(\)content\(\)
       \lceil \langle key = val \rangle \rceil
                                                                                                  \lceil \langle key = val \rangle \rceil
          \item \langle item content \rangle
                                                                                                      \item \langle item content \rangle
          \item [\langle custom \rangle] \langle item content \rangle
                                                                                                      \item [\langle custom \rangle] \langle item content \rangle
          \forall item^* \langle item content \rangle
                                                                                                      \item* \(\(\)item \(\)content\)
          \forall item^* [\langle content \rangle] \langle item content \rangle
                                                                                                     \forall item^*[\langle content \rangle] \langle item content \rangle
       \end{keyans}
                                                                                                  \end{keyans*}
\end{enumext}
                                                                                           \end{enumext}
```

The $\langle keys \rangle$ set in the *optional argument* of the environment are the same (almost) as those of the enumext and enumext* environments and have *higher precedence* than those set by \setenumext[$\langle keyans \rangle$] { $\langle key = val \rangle$ } or \setenumext[$\langle keyans^* \rangle$] { $\langle key = val \rangle$ }. If the *optional argument* is not passed or the $\langle keys \rangle$ are not set by \setenumext, the default values will be the same as the "second level" of the enumext environment with the difference in the $\langle label \rangle$ which will be set to label=\Alph*).

6.4.1 The \item* in keyans and keyans*

\item* \item*

```
\item* [\langle content \rangle]
```

The \item* and \item* [$\langle content \rangle$] command "store" the current $\langle label \rangle$ set by label key next to the optional argument $\langle content \rangle$ in sequence and prop list { $\langle store\ name \rangle$ } set by save-ans key in the "first level" of the enumext or enumext* environments.

The *starred argument* '*' cannot be separated by spaces '\(\)' from the command, i.e. \item* and the *optional argument* does "NOT" support *verbatim content*. By design it is assumed that the \item* will only appear "once" within the environment.

The behavior of \item* in keyans and keyans* environments is NOT the same as in the enumext or enumext* environments.

Example

```
\begin{enumext}[save-ans=test,columns=2,show-ans=true]
  \item Text containing a question.
    \begin{keyans*}[nosep,columns=2]
      \item Choice
      \item* Correct choice
      \item Choice
      \item Choice
      \item Choice
    \end{keyans*}
  \item Text containing a question and image.
    \begin{keyans}[nosep,mini-env={0.4\linewidth}]
      \item Choice
      \item Choice
      \item Choice
      \item Choice
      \times_{note} \ Correct choice
      \miniright
      \includegraphics[scale=0.25]{example-image-a}
      Some text
    \end{keyans}
\end{enumext}
```

- 1. Text containing a question.
 - A) Choice
- * B) Correct choice
- C) Choice E) Choice
- D) Choice
- 2. Text containing a question and image.
 - A) Choice
- B) Choice
- C) Choice
- D) Choice
- * E) [note] Correct choice



Some text

The environment keyanspic

keyanspic \begin{keyanspic} $[\langle key = val \rangle]$ \anspic* $[\langle content \rangle]$ \def drawing or tabular\range} \begin{keyanspic}

The keyanspic environment is an "enumerated list" environment activated by the save-ans key that has the same configuration for "spacing" and $\langle label \rangle$ as the keyans environment that uses the \anspic command instead of \item. It is intended for placing drawings or tabular with $\langle label \rangle$ centered above or below in a single line or upper and lower layout style. A representation of the output can be seen in the figure 6.

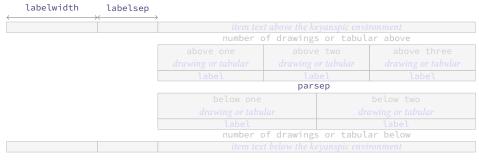


Figure 6: Representation of the keyanspic environment with layout-sty= $\{\langle 3, 2 \rangle\}$ in enumext.

When the keyanspic environment is used without keys the \langle labels \rangle are centered below the drawings or tabular in a *single line* layout style.

This environment cannot be nested and must always be at the "first level" of the enumext environment, the \item command is disabled and keys cannot be set using \setenumext.

6.5.1 Keys for keyanspic

label-pos = $\{\langle above \mid below \rangle\}$

default: below

Set the position of \(\lambda label\rangle\) to be centered "above" or "below" drawings or tabular when the \anspic command is executed.

 $label-sep = \{\langle rubber \ length \ | \ rigid \ length \rangle\}$

default: internal adjustment

Set the vertical spacing between the \langle label \rangle centered "above" or "below" and drawings or tabular when running the \anspic command.

 $layout-sty = \{ \langle n^{\circ} upper, n^{\circ} lower \rangle \}$

Set the number of drawings or tabular that will be distributed "upper" and "lower" within the environment when executing the \anspic command. The value must be passed in braces and if not set or the $\langle n^o | lower \rangle$ is omitted the *drawings* or *tabular* will be put on a *single line*.

```
\texttt{layout-sep} = \{ \left< \textit{rubber length} \mid \textit{rigid length} \right> \}
```

default: adjusted parsep from keyans

Set the vertical separation between the number of drawings or tabular placed at the "upper" and "lower" within the environment when executing the \anspic command. Internally adjusts the parsep value taken from the keyans environment.

```
layout-top = \{ \langle rubber \ length \ | \ rigid \ length \rangle \}
```

default: adjusted topsep from keyans

Set the vertical space added to both the top and bottom of the environment. Internally adjust the value of topsep taken from keyans environment.

6.5.2 The command \anspic

```
\anspic \anspic{\langle drawing \ or \ tabular \rangle}
                \arrowvert anspic*[\langle content \rangle] \{\langle drawing \ or \ tabular \rangle\}
```

The \anspic command take three arguments, the *starred argument* '*' store the current $\langle label \rangle$ next to the optional argument (content) in sequence and prop list {(store name)} set by save-ans key.

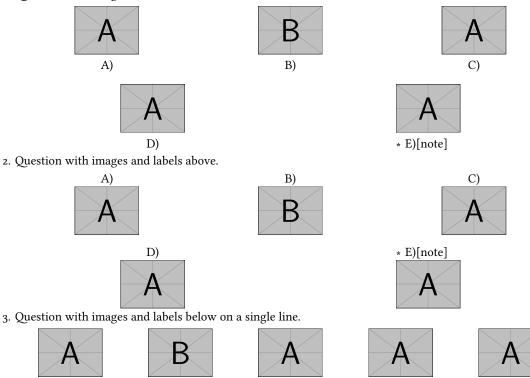
The starred argument '*' cannot be separated by spaces 'u' from the command, i.e. \anspic* and the optional argument does "NOT" support verbatim content. By design it is assumed that the starred argument '*' will only appear "once" within the environment.

Example

```
\begin{enumext}[save-ans=test,show-ans,nosep]
  \item Question with images and labels below.
    \begin{keyanspic}[layout-sty={3,2}]
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic{\includegraphics[scale=0.15]{example-image-b}}
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic*[note]{\includegraphics[scale=0.15]{example-image-a}}
    \end{keyanspic}
```

```
\item Question with images and labels above.
    \begin{keyanspic}[label-pos=above, layout-sty={3,2}]
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic{\includegraphics[scale=0.15]{example-image-b}}
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic*[note]{\includegraphics[scale=0.15]{example-image-a}}
    \end{keyanspic}
  \item Question with images and labels below on a single line.
    \begin{keyanspic}
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic{\includegraphics[scale=0.15]{example-image-b}}
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic{\includegraphics[scale=0.15]{example-image-a}}
      \anspic*[note]{\includegraphics[scale=0.15]{example-image-a}}
    \end{keyanspic}
\end{enumext}
```

1. Question with images and labels below.



💞 Preferably use label-pos=above when creating a tagged PDF, this will preserve the reading order and navigation of the document.

C)

D)

* E)[note]

6.6 **Printing stored content**

B)

The command \getkeyans

\getkeyans \getkeyans{\langle store name: position\rangle}

A)

The command \getkeyans prints the "stored content" in prop list {\store name\}\ defined by save-ans key in the $\langle position \rangle$ returned by the show-pos key. The "stored content" can only be accessed after it is stored, if $\{\langle store\ name \rangle\}\$ does not exist the command will return an error.

The form taken by the argument {\(\store \ name : position\)\)} is the same as that used to generate the "internal label and ref" system when save-ref key are active, so to refer to a "stored content". For example \getkeyans{test:4} will return the "stored content" at position 4 of the environment in which the key save-ans=test was set.

6.6.2 The command \foreachkeyans

 $\foreachkeyans \foreachkeyans \{\langle key = val \rangle\} \} \{\langle store \ name \rangle\}$

The command \foreachkeyans goes through and executes the command \getkeyans on the contents in prop *list* { $\langle store\ name \rangle$ }. If you pass without options run \getkeyans on all contents in *prop* list { $\langle store\ name \rangle$ }.

Options for command

```
sep = \{\langle code \rangle\}
                                                                                                                                                                                             default: {; }
```

Establishes the *separation* between "each" $\{\langle content \rangle\}$ stored in *prop list* $\{\langle store\ name \rangle\}$. For example, you can use $sep=\{\{\{10pt]\}\}$ for vertical separation of stored contents.

```
step = \{\langle integer \rangle\}
```

Sets the *step* (increment) applied to the value set by key start for each $\{\langle content \rangle\}$ stored in *prop list* $\{\langle store \rangle\}$ name}. The value must be a $\langle positive integer \rangle$.

Sets the position of the prop list {\store name\} from which execution will start. The value must be a \positive integer\.

```
stop = \{\langle integer \rangle\}
                                                                                                                                                                                    default: 0
```

Sets the position of the prop list {\store name}} from which execution it will finish executing. The value must be a *\(\positive integer\)*.

```
before = \{\langle code \rangle\}
                                                                                                                                                               default: emptv
```

Sets the $\{\langle code \rangle\}$ that will be executed $\langle before \rangle$ each $\{\langle content \rangle\}$ stored in *prop list* $\{\langle store\ name \rangle\}$. The $\{\langle code \rangle\}$ must be passed between braces.

```
after = \{\langle code \rangle\}
                                                                                                                                                                 default: empty
```

Sets the $\{\langle code \rangle\}$ that will be executed $\langle after \rangle$ each $\{\langle content \rangle\}$ stored in *prop list* $\{\langle store\ name \rangle\}$. The $\{\langle code \rangle\}$ must be passed between braces.

```
wrapper = \{ \langle code \{ #1 \} \ more \ code \rangle \}
                                                                                                                                                                   default: empty
```

Wraps the $\{\langle content \rangle\}$ stored in prop list $\{\langle store\ name \rangle\}$ referenced by $\{\#1\}$. The $\{\langle code \rangle\}$ must be passed between braces. For example $\foreachkeyans[wrapper={\mbox{$\mbox{[1em][1]}$}}]{\mbox{$\mbox{$\mbox{$\mbox{$}\mbox{$}\mbox{$}$}}}.$

6.6.3 The command \printkeyans

```
\printkeyans \printkeyans{\(\store\) name\)}
                       \printkeyans[\langle keys \rangle] \{\langle store\ name \rangle\}
                       \printkeyans*[\langle keys \rangle] \{\langle store\ name \rangle\}
```

The command \printkeyans prints "all stored content" in sequence {\store name\ranger} defined by save-ans key placing this inside the enumext environment by default or the enumext* environment if the starred argument '*' is used.

The "stored content" can only be accessed after it is stored in the sequence, if { (store name) } does not exist the command will return an error.

The optional argument allows managing the \(\lambda \text{keys} \) in the "first level" of the environment in which the "stored content" of the sequence {\store name\} will be printed, if the starred argument '*' is used it will be enumext*

The default values for the "first level" are the same as the default values for the enumext and enumext* environments along with the keys nosep, first=\small, font=\small and columns=2. For the inner levels of the environment enumext saved in the sequence { (store name)} the default values are the same as those established for the second, third and fourth levels plus the keys nosep, first=\small, font=\small. If the environment enumext* is saved within the sequence $\{\langle store\ name \rangle\}$ it will have the same default values plus the keys nosep, first=\small, font=\small.

Since the command encapsulates by default the enumext environment or the enumext* environment, we must take some considerations:

- If we execute \printkeyans*{\store name\sequence {\store name\sequence} already contains any enumext* environment an error will be returned as we cannot nest.
- If we execute \printkeyans*{\(\store name \)\)} and the sequence {\(\store name \)\)} contains any enumext environments, they will start with the $\langle keys \rangle$ set for the first level unless they are set in the optional argument or save-key is used to modify it.
- If we execute $\printkeyans{\langle store\ name \rangle}$ and the sequence $\{\langle store\ name \rangle\}$ contains any environment enumext*, they will start with the $\langle keys \rangle$ set by default unless they are set in the *optional argument* or save-key is used to modify it.

The default values for the "first level" of \printkeyans commands and \printkeyans* are established using \setenumext[$\langle print, 1 \rangle$] { $\langle keys \rangle$ } and \setenumext[$\langle print^* \rangle$] { $\langle keys \rangle$ }.

If we need to set the $\langle keys \rangle$ for the environment enumext "saved" in the sequence $\{\langle store\ name \rangle\}$ we will use \setenumext[$\langle print, level \rangle$] { $\langle keys \rangle$ } and if we need to set the $\langle keys \rangle$ for the environment enumext* "saved" in the sequence $\{\langle store\ name \rangle\}\$ we will use $\$ setenumext $[\langle print, * \rangle]$ $\{\langle keys \rangle\}$.

Example

```
\begin{enumext} [save-ans=sample,columns=1,show-pos=true,nosep,save-ref=true]
  \item Factor 3x+3y+3z. \anskey\{3(x+y+z)
  \item True False
    \begin{enumext}[nosep]
      \item \LaTeX2e\ is cool? \anskey{Very True!}
    \end{enumext}
  \item Related to Linux
    \begin{enumext}[nosep]
      \item You use linux? \anskey{Yes}
      \item Rate the following package and class
        \begin{enumext}[nosep]
          \item \texttt{xsim} \anskey{very good}
          \item \texttt{exsheets} \anskey{obsolete}
        \end{enumext}
    \end{enumext}
\end{enumext}
The answer to \ref{sample:4} is \getkeyans{sample:4} and the answers to
all the worksheets are as follows:
\printkeyans{sample}
```

- 1. Factor 3x + 3y + 3z.
- [1] | 3(x+y+z)
- 2. True False
 - (a) LATEX2e is cool?
 - [2] Very True!
- 3. Related to Linux
 - (a) You use linux?
 - [3] Yes
 - (b) Rate the following package and class
 - xsim
 - [4] very good
 - exsheets ii.
 - [5] obsolete

The answer to 3.(b).i is very good and the answers to all the worksheets are as follows:

```
1. 3(x+y+z)
2. (a) Very True!
3. (a) Yes
  (b) i. very good
         obsolete
```

Full examples

Here I will leave as an example some adaptations questions taken from TeX-SX. The examples are attached to this documentation and can be extracted from your PDF viewer or from the command line by running:

```
$ pdfdetach -saveall enumext.pdf
```

and then you can use the excellent arara1 tool to compile them.

Example 1

Adapted from the response given by Enrico Gregorio in Squares for answer choice options and perfect alignment to mathematical answers

- 1. La velocità di $1{,}00 \times 10^2 \,\mathrm{m/s}$ espressa in km/h è:
 - A 36 km/h.
 - B 360 km/h.
 - C 27,8 km/h.
 - D $3,60 \times 10^8 \,\text{km/h}$.

- $10^{-15}\,\mathrm{m}$). Qual è la relazione tra queste due unità di
- A $1 \text{ Å} = 1 \times 10^5 \text{ fm}.$
- B $1 \text{ Å} = 1 \times 10^{-5} \text{ fm}.$
- C 1 Å = 1 × 10⁻¹⁵ fm.
- D $1 \text{ Å} = 1 \times 10^3 \text{ fm}.$
- 2. In fisica nucleare si usa l'angstrom (simbolo: $1 \, \text{Å} =$ 1×10^{-10} m) e il fermi o femtometro (1 fm = 1×3 . La velocità di $1,00 \times 10^2$ m/s espressa in km/h è:

©2024 by Pablo González L 19/156

¹The cool T_EX automation tool: https://www.ctan.org/pkg/arara

- A 36 km/h.
- $360 \,\mathrm{km/h}$.
- $27.8 \, \text{km/h}$.
- $D = 3,60 \times 10^8 \, \text{km/h}.$
- 4. In fisica nucleare si usa l'angstrom (simbolo: 1 Å = 1×10^{-10} m) e il fermi o femtometro (1 fm = $1 \times$

2. A

 $10^{-15}\,\mathrm{m}$). Qual è la relazione tra queste due unità di misura?

4. A

 $10^{-15}\,\mathrm{m}$). Qual è la relazione tra queste due unità di

- A $1 \text{ Å} = 1 \times 10^5 \text{ fm}.$
- $\overline{\text{B}} \, 1 \, \text{Å} = 1 \times 10^{-5} \, \text{fm}.$
- $\boxed{\text{C}} 1 \text{ Å} = 1 \times 10^{-15} \text{ fm}.$
- D $1 \text{ Å} = 1 \times 10^3 \text{ fm}.$

1. B Example 2

Adapted from the response given by Florent Rougon in Multiple choice questions with proposed answers in random order — addition of automatic correction (cross mark)

3. B

- 1. La velocità di $1,00 \times 10^2$ m/s espressa in km/h è: 3. La velocità di $1,00 \times 10^2$ m/s espressa in km/h è:
 - A 36 km/h.
- ✓ B 360 km/h.
 - C 27,8 km/h.
 - D $3,60 \times 10^8 \,\text{km/h}$.

 \checkmark A 1 Å = 1 × 10⁵ fm.

B $1 \text{ Å} = 1 \times 10^{-5} \text{ fm}.$

 $C 1 Å = 1 \times 10^{-15} \text{ fm}.$

- A 36 km/h.
- ✓ B 360 km/h.

misura?

- C 27,8 km/h.
- D $3,60 \times 10^8 \,\text{km/h}$.
- 2. In fisica nucleare si usa l'angstrom (simbolo: 1 Å = 4. In fisica nucleare si usa l'angstrom (simbolo: 1 Å = 4) 1×10^{-10} m) e il fermi o femtometro (1 fm = 1×10^{-10} m) e il femtometro (1 fm = 1×10^{-10} m) e il femtometro (1 fm = 1×10^{-10} m) e il femtometro (1 fm = 1×10^{-10} m) e il femtometro (1 fm = 1×10^{-10} m) e il femtometro (1 fm = 1×10^{-10} m) e il femtometro ($10^{-15}\,\mathrm{m}$). Qual è la relazione tra queste due unità di misura?
- \checkmark A | 1Å = 1 × 10⁵ fm.
 - B $1 \text{ Å} = 1 \times 10^{-5} \text{ fm}.$
 - $C 1 Å = 1 \times 10^{-15} \text{ fm}.$
 - D $1 \text{ Å} = 1 \times 10^3 \text{ fm}.$

- D $1 \text{ Å} = 1 \times 10^3 \text{ fm}.$ 1. B
- 3. B

※ 2. A * 4. A

×

Example 3

A "simple multiple choice" test .

- 1. First type of questions
 - A) value
 - C) value

- B) correct
- D) value

(D) I and III only

(E) I, II, and III

- 2. Second type of questions
 - I. $2\alpha + 2\delta = 90^{\circ}$
 - II. $\alpha = \delta$
 - III. $\angle EDF = 45^{\circ}$
 - (A) I only
 - (B) II only
 - © I and II only
- 3. Third type of questions
 - (1) $2\alpha + 2\delta = 90^{\circ}$
 - (2) $\angle EDF = 45^{\circ}$
 - (A) value
 - (B) value
 - (C) value

- (D) value
- (E) value
- 4. Question with image and label below:











- 5. Question with image on left side:
 - (A) value
 - (B) value
 - (C) value
 - (D) correct
- (E) value
- Test keys
- 2. D
- 3. C, some note





- 1. B), x = 5

- * 4. E, A duck
- * 5. D, other note

Example 4

A "simple worksheet" using ducks :) 🖹



Factor $x^2 - 2x + 1$



Factor 3x + 3y + 3z

The following questions need to be cuaqtified:)



True False

- (a) $\alpha > \delta$
- (b) LaTeX2e is cool?



Related to Linux

- (a) You use linux?
- (b) Usually uses the package manager?
- (c) Rate the following package and class
 - i. xsim-exam
 - ii. xsim
 - iii. exsheets

The answer to 1 is $(x-1)^2$ and the answer to 3.(a) is False.

- 1. $(x-1)^2$
- 2. 3(x+y+z)
- 3. (a) False
 - (b) Very True!
- 4. (a) Yes

(b) Yes, dnf

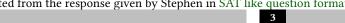
×.

×

- (c) i. doesn't exist for now :(
 - very good
 - iii. obsolete

Example 5

Adapted from the response given by Stephen in SAT like question format 🖹



Which choice best describes what happens in the passage?

- A) One character argues with another character who intrudes on her home.
- One character receives a surprising request from another character.
- C) One character reminisces about choices she has made over the years.
- D) One character criticizes another character for pursuing an unexpected course of action.

Which choice best describes what happens in the passage?

- A) One character argues with another character who intrudes on her home.
- One character receives a surprising request from another character.
- One character reminisces about choices she has made over the years.
- D) One character criticizes another character for pursuing an unexpected course of action.

Which choice best describes what happens in the

- A) One character argues with another character who intrudes on her home.
- B) One character receives a surprising request from another character.
- C) One character reminisces about choices she has made over the years.
- D) One character criticizes another character for pursuing an unexpected course of action.

Which choice best describes what happens in the passage?

- A) One character argues with another character who intrudes on her home.
- One character receives a surprising request from another character.
- C) One character reminisces about choices she has made over the years.
- D) One character criticizes another character for pursuing an unexpected course of action.

1. A)

2. C)

3. B)

4. D)

Tagged PDF examples

This section is just to show the compatibility of enumext with tagged PDF using lualatex. The attached files here are just for testing and are intended as examples and, in a way, to simplify the time of Matthew Bertucci (@mbertucci) when he sees this excellent package and adds it to The LaTeX Tagged PDF repository.

To compile the tests with lualatex-dev the packages multicol, scontents, unicode-math, geometry, graphicx and hyperref are required.

- The file enumext-01. tex contains the basic tests for the enumext and enumet* environments and the nesting between them plus the use of the label, labelwidth, labelsep, ref, align and wrap-label keys. Source file and tagged PDF .
- The file enumext-02.tex contains the tests for the enumext and enumet* environments and the support for minipage and multicols environments using the keys columns, columns-sep, mini-env, miniright and \miniright command. Source file and tagged PDF \(\frac{1}{2} \).

- The file enumext-03.tex contains the tests for the enumext and keyanspic environments activated by the save-ans key together with the save-sep and save-ref keys and the \printkeyans command. Source file and tagged PDF .
- The file <code>enumext-04.tex</code> contains the tests for the <code>\anskey</code> command and the <code>anskey*</code> environment activated by the <code>save-ans</code> key along with the <code>\getkeyans</code> and <code>\printkeyans</code> commands. Source file <code>and tagged PDF</code> <code>\textstyle{L}</code>.
- The file enumext-05.tex contains the tests for the environments keyans, keyans* and keyanspic activated by the key save-ans together with the keys no-store and show-ans and the commands \setenumext, \setenumextmeta, \printkeyans and \foreachkeyans. Source file and tagged PDF .

9 The way of non-enumerated lists

It is possible to use (or abuse) the enumext environment to mimic *non-enumerated* list environments such as itemize and description, clearly the $\langle keys \rangle$ to "store answers", the keyans and keyanspic environments lose their sense and it is not the focus of the main of this package, but, why not to do it?.

Here I leave as an example other uses of the enumext environment that can be helpful for specific purposes. The "trick" to generate these fake environments is set label= $\{\}$ or label= $\{\langle some \rangle\}$ and play with the list-indent, list-offset, font and wrap-label keys.

Fake itemize environment

Here we set the label key using the default settings in MTeX for the four levels \textbullet, \textendash, \textasteriskcentered and \textperiodcentered together with the nosep key to reduce the vertical spaces in the left side example and set the label key in mathematical mode for the right side as \ast, \diamond, \circ and \star for the four levels together with the nosep key

- First level item
 - Second level item
 - * Third level item
 - · Fourth level item
- First level item

- * First level item
 - ♦ Second level item
 - Third level item
 - ⋆ Fourth level item
- * First level item

Fake description environment

Here we set label={} and list-indent=2.5em, font=\bfseries.

SomeThing A short one-line description.

This is an entry without a label.

 $\textbf{Something} \ \textbf{A} \ \textbf{short} \ \textit{one-line} \ \textbf{description} \ \textbf{text}.$

Something long A much *longer* description text may take more than one line or more than one paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

If we add list-indent=Opt you get widest style:

SomeThing A short one-line description.

This is an entry without a label.

Something A short *one-line* description text.

Something long A much *longer* description text may take more than one line or more than one paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

The small space at the beginning of the "unlabeled entry" corresponds to \labelsep and can be removed using \hspace{-\labelsep} at the beginning of the line.

Description indented by label

Here we set label={} and we will give a convenient value to labelsep and labelwidth, for example we can take as reference our *longest label* and pass it as value using:

```
\newlength{\descitemwd}
\settowidth{\descitemwd}{\textbf{Something long}}}
```

and then use labelsep=4pt, labelwidth=\descitemwd, font=\bfseries.

SomeThing A short one-line description.

This is an entry without a label.

Something A short one-line description.

Something long A much longer description. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

The environment can be translated so that the $\langle labels \rangle$ are on the left margin calculating the value passed to the list-offset key, in this case it will be equal to the sum of the values set by the labelwidth and labelsep keys finally resulting as list-offset={-\descitemwd - 4pt}.

SomeThing A short one-line description.

This is an entry without a label.

Something

A short one-line description.

Something long A much longer description. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum

ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

If we add align=right it will look like this:

SomeThing A short one-line description.

This is an entry without a label.

Something A short one-line description.

Something long A much longer description. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

At this point we have used list-offset={-\descitemwd - 4pt} instead of list-offset={-\labelwidth - \labelsep}, this is because the parameters \labelwidth and \labelsep take the default values, as if we had not set label.

Description with multi-line labels

The label key does not accept *multiline material*, this is where the wrap-label and wrap-label* keys comes into play. Unlike the enumitem package, the align key only supports three options, so what we will do is create a command in the style \parleft of enumitem that allows us to place *multiline labels* using \parbox.

Now we just need to set wrap-label*={\labelbx{#1}}.

SomeThing A short one-line description.

This is an entry without a label.

Something A short one-line description.

Something A much longer description. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum **long** ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

SoMeThInG A much longer description. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum **LoNg** ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Final notes

The original implementation (if you can call it that) of the ideas that led to the creation of enumext were some macros using the enumerate[5] package for personal use created in early 2003, the code was quite questionable, but functional for these simple requirements.

With the great answers given by Christian Hupfer in Create a fake label ref using list and the answer given by David Carlisle in Change the use of label ref by data save in an array (list) I managed to create a more solid code than the original version, now using the <code>l3prop[11]</code> and <code>l3seq[11]</code> modules together with the <code>hyperref[8]</code> and <code>enumitem[6]</code> packages, which did the job, but with some limitations.

As time went by I took these limitations as a personal challenge which I called "reinventing the wheel", since there were packages and classes that did more or less what I was looking for, but did not fit my simple requirements. This "reinventing the wheel" finally ended up becoming enumext.

Why list environments?

The answer is simple, first I love the beauty of its syntax and many of what I had already written used the enumerate environment or lists created using the enumitem package. In my mind I thought: how complicated could it be to write a package that looked like enumitem? It seemed simple enough, of course I didn't have in mind the mess I was getting into working with list environments, minipage and adding support for the multicol and hyperref packages.

Of course, seeing the final result of the experiment "reinventing the wheel" I am quite satisfied.

Why not random questions and other utilities

The "random" type questions I love and hate them at the same time, although they simplify a lot the work when creating a multiple choice test, but you lose the beauty of typessetting a document with ETEX, that is to say the output does not always look as nice as it should, even if they are only alternatives these must follow a certain order when presented either numerical or presentation, that said handling that using *nested lists* is quite complicated so I do not classify to be implemented.

©2024 by Pablo González L 23/156

Why has it taken so long?

One of the setbacks, beyond my laziness, was including compatibility with *tagged* PDF. To be honest, it's something I never considered at any point, but I firmly believe that being able to create *accessible documents* provides a great opportunity in the world of mathematics education. From my perspective as a *high school* teacher, beyond theorems and deep mathematics, the use of exercise lists is one of the most common things. Being able to open the way to work in parallel with those who have different abilities is really important and I regret not having looked into this in the past. I hope that enumext serves this purpose and inspires more users and authors to follow this path.

10 References

- [1] HIRSCHHORN, PHILIP. "Using the exam document class". Available from CTAN, https://www.ctan.org/pkg/exam, 2023.
- [2] NIEDERBERGER, CLEMENS. "xsim eXercise Sheets IMproved". Available from CTAN, https://www.ctan.org/pkg/xsim, 2023.
- [3] MITTELBACH, FRANK. "An environment for multicolumn output". Available from CTAN, https://www.ctan.org/pkg/multicol, 2024.
- [4] González, Pablo. "scontents Stores LTEX contents in memory or files". Available from CTAN, https://www.ctan.org/pkg/scontents, 2024.
- [5] The LTEX Project. "enumerate Enumerate with redefinable labels". Available from CTAN, https://www.ctan.org/pkg/enumerate, 2024.
- [6] Bezos, Javier. "Customizing lists with the enumitem package". Available from CTAN, https://www.ctan.org/pkg/enumitem, 2019
- [7] BERRY, KARL. "ΜΈΣΧ 2_ε: An Unofficial Reference Manual". Available from ctan, https://ctan.org/pkg/latex2e-help-texinfo, 2024.
- [8] The LTEX Project. "Extensive support for hypertext in LTEX". Available from CTAN, https://www.ctan.org/pkg/hyperref, 2024.
- [9] Burnol, Jean-François. "The footnotehyper package". Available from ctan, https://www.ctan.org/pkg/footnotehyper, 2021.
- [10] The LTEX Project. "The expl3 package". Available from CTAN, https://www.ctan.org/pkg/l3kernel, 2024.
- [11] The LTEX Project. "The LTEX3 Interfaces". Available from CTAN, https://www.ctan.org/pkg/l3kernel, 2024.
- [12] The LTEX Project. "The LTEX 2_{ε} sources". Available from CTAN, https://ctan.org/tex-archive/macros/latex/base, 2024.
- [13] The LTeX Project. "LTeX for authors current version". Available from CTAN, https://ctan.org/pkg/latex-base, 2024.
- [14] Gundlach, Patrick. "The lua-visual-debug package". Available from ctan, https://www.ctan.org/pkg/lua-visual-debug, 2023.
- [15] Lemvig, Mogens. "The shortlst package". Available from ctan, https://www.ctan.org/pkg/shortlst, 1998.
- [16] NIEDERBERGER, CLEMENS. "tasks Horizontally columned lists". Available from CTAN, https://www.ctan.org/pkg/tasks, 2022.

11 Change history

v1.0 2024-10-06 - First public release.

©2024 by Pablo González L 24/156

12 Index of Documentation

The italic numbers denote the pages where the corresponding entry is described.

C	F
Document class:	\footnote 5
article 2	•
book	I
exam 2	\itemsep 8
letter 2	K
report 2	Keys for \anskey provide by enumext:
\columnbreak 4, 13	break-col
Commands provide by enumext:	item-join
\anskey 11-14	item-pos* 13
\anspic	item-star
\foreachkeyans	item-sym* 13
\getkeyans	Keys for \foreachkeyans provide by enumext:
\item* 5-7, 11-13, 15	after 18
\item 5-7, 10, 11, 13, 15, 16	before
\miniright	sep
\printkeyans 6, 12, 18	start
\setenumextmeta 6	step
\setenumext 5-7, 11, 12, 15, 18	wrapper
Counters defined by enumext: enumXiii 4	Keys for anskey* provide by enumext:
enumXii 4	break-col
enumXiv 4	force-eol 14
enumXi 4	item-join
enumXviii 4	item-pos* 13
enumXvii 4	item-star 13
enumXvi 4	item-sym* 13
enumXv 4	overwrite 14
.	write-env
E Environmente provide by enumert	Keys for environments provide by enumext: above*
Environments provide by enumext: anskey*	above 9
enumet* 21	after
enumext* 4-15, 18	align
enumext 4-16, 18, 21, 22	base-fix 8
kenyans* 9	before* 9, 10
keyans* 4-15, 22	before 9
keyanspic 4, 7, 8, 11–14, 16, 22	below* 9
keyans 4-16, 22	below 9
Environments:	check-ans
Verbatim	columns-sep 4, 10, 21
center 5	columns 4, 9, 10, 21 first
description 5, 22 enumerate	fort 7
figure	item-pos*
flushleft 5	item-sym* 5, 6
flushright 5	itemindent 9, 10
itemize 5, 22	itemsep 8
keyans 22	label-indent 9
list 3, 5, 9, 23	label-pos
minipage 3-5, 8-11, 21, 23	label-sep
multicols	labelsep 3-7, 9, 10, 12, 21, 22
quotation 5	labelwidth 3, 4, 6, 7, 9, 10, 12, 21, 22
quote 5 tabbing 5	label
table 5	labewdith
task 5	layout-sep
trivlist 5	layout-sty
verbatim 5	layout-top
verse 5	list-indent 3, 9
©2024 by Pablo González L	

list-offset 3, 9, 22, 23	Labels provide by enumext:
listparindent 9, 10	\Alph* 7, 8, 15
mark-ans 12	\Roman*
mark-pos 12	\alph*, 7, 8
mark-ref 12	\arabic*
mini-env 4, 9, 11, 21	\roman*
mini-right* 7, 11	\labelsep
mini-right 7, 11, 21	\labelwidth
mini-sep	\linewidth 12
mode-box 7	\listparindent 9
no-store 11, 13, 14, 22	
noitemsep 8	P
nosep 8, 22	Packages:
overwrite 14	enumerate 23
parsep 8, 10, 16	enumext 1-5, 7, 16, 21, 23, 22
partopsep	enumitem 3-5, 23
ref 4, 8, 21	fancyvrb
resume* 7, 10, 11	footnotehyper
resume 7, 10, 11	geometry 22
rightmargin 9	graphicx 23
save-ans 4, 6, 10–18, 22	hyperref
save-key 10–12, 18	l3keys
save-ref 4, 7, 12-14, 17, 22	l3prop
save-sep 12, 22	l3seq
series 7, 10, 11	multicol
show-ans 12, 22	scontents
show-length 8	task
show-pos	unicode-math 22
start* 10	xsim 2
start 10	\parsep 8
topsep 8, 9, 16	\partopsep {
widest 7	
wrap-ans	R
wrap-label*	\raggedcolumns
wrap-label 7, 21, 23	\ref
wrap-opt	\rightmargin
write-env 14	
L	Т
hel	\tonsen

©2024 by Pablo González L 26/156

13 Implementation

The most recent publicly released version of enumext is available at CTAN: https://www.ctan.org/pkg/enumext. While general feedback via email is welcomed, specific bugs or feature requests should be reported through the issue tracker: Ohttps://github.com/pablgonz/enumext/issues.

The documentation presented here is far from professional, it contains a lot of obvious information that to the eye of a TeXpert are superfluous, but, after so many years developing this project is the only way to remember what does what.

13.1 General conventions

Variables containing i, ii, iii and iv are associated by level with the enumext environment, variables containing v are associated with the keyans environment, variables containing vi are associated with the keyanspic environment, variables containing vii are associated with the enumext* environment and variables containing viii are associated with the keyans* environment.

To simplify writing and documentation some variables and functions that are common to the different levels of the environments are described using a capital "X".

The temporary function __enumext_tmp:n is used in different parts of the package code for variable creation or execution of other functions that are grouped into this one.

All variables and functions defined in this package are private and are NOT intended to work or be used by another package or module.

13.2 Initial set up

Start the DocStrip guards.

```
*package
```

Identify the internal prefix (FTFX3 DocStrip convention) for l3doc class.

```
2 (@@=enumext)
```

13.3 Declaration of the package

First we will make sure we have a minimum (super updated) version of LTFX to work correctly.

```
3 \NeedsTeXFormat{LaTeX2e} [2024-06-01]
```

Now declare the enumext package.

```
4 \ProvidesExplPackage
5 {enumext}
6 {2024-10-06}
7 {1.0}
8 {Enumerate exercise sheets}
```

Finally check if the multicol and scontents packages are loaded, if not we load it.

```
9 \hook_gput_code:nnn {begindocument} {enumext}
      \IfPackageLoadedTF { multicol }
        {
          \msg_info:nnn { enumext } { package-load } { multicol }
        }
        {
          \msg_info:nnn { enumext } { package-not-load } { multicol }
          \RequirePackage{multicol}[2024-05-23]
18
        }
      \IfPackageLoadedTF { scontents }
19
        {
          \msg_info:nnn { enumext } { package-load } { scontents }
21
        }
        {
23
          \msg_info:nnn { enumext } { package-not-load } { scontents }
24
          \RequirePackage{scontents}
25
    }
```

13.4 Definition of variables

Variables that do not appear in this section are created by means of \keys_define:nn or some function described below.

```
\l__enumext_level_int Integer variables will control the nesting levels of the environments and \anskey command.
     \l__enumext_level_h_int
                                 28 \int_new:N \l__enumext_level_int
\l__enumext_anskey_level_int
                                 29 \int_new:N \l__enumext_level_h_int
\l__enumext_keyans_level_int
                                30 \int_new:N \l__enumext_anskey_level_int
                                int_new:N \l__enumext_keyans_level_int
      \l__enumext_keyans_level_h_int
                                32 \int_new:N \l__enumext_keyans_level_h_int
     \l__enumext_keyans_pic_level_int
                                 33 \int_new:N \l__enumext_keyans_pic_level_int
                                (End of definition for \l_enumert_level_int and others.)
                                Internal variables used by functions \__enumext_is_not_nested:, \__enumext_is_on_first_level:
    \l enumext starred bool
    \g__enumext_starred_bool
                                and \__enumext_keyans_name_and_start: (§13.5.1).
      \l__enumext_starred_first_bool
                                 34 \bool_new:N \l__enumext_starred_bool
    \l__enumext_standar_bool
                                 35 \bool_new:N \g__enumext_starred_bool
                                36 \bool_new:N \l__enumext_starred_first_bool
    \g__enumext_standar_bool
                                37 \bool_new:N \l__enumext_standar_bool
      \l__enumext_standar_first_bool
                                38 \bool_new:N \g__enumext_standar_bool
 \l__enumext_anskey_env_bool
                                39 \bool_new:N \l__enumext_standar_first_bool
 \l__enumext_keyans_env_bool
                                40 \bool_new:N \l__enumext_anskey_env_bool
   \g__enumext_start_line_tl
                                \bool_new:N \l__enumext_keyans_env_bool
   \g__enumext_envir_name_tl
                                 42 \tl_new:N \g__enumext_start_line_tl
   \l__enumext_envir_name_tl
                                 43 \tl_new:N \g__enumext_envir_name_tl
                                 44 \tl_new:N \l__enumext_envir_name_tl
                               (End of definition for \l_enumert_starred_bool and others.)
                               Variables to store the "name of the counters" enumXi, enumXii, enumXiii and enumXiv for enumext en-
    \l__enumext_counter_i_tl
                               vironment, enumXv for keyans environment and enumXvi for the keyanspic environment. The counters
   \l__enumext_counter_ii_tl
  \l__enumext_counter_iii_tl
                                enumXvii and enumXviii are used by enumext* and keyans* environments.
   \l__enumext_counter_iv_tl
                               The initial values of these variables are set by the function \__enumext_define_counters: Nn (§13.10) and
    \l__enumext_counter_v_tl
                                then modified by the function \__enumext_label_style: Nnn used by label key (§13.13).
   \l__enumext_counter_vi_tl
                                 45 \cs_set_protected:Npn \__enumext_tmp:n #1
  \l enumext counter vii tl
 \l__enumext_counter_viii_tl
                                       \tl_new:c { l__enumext_counter_#1_tl }
                                 47
                                    7
                                 49 \clist_map_inline:nn { i, ii, iii, iv, v, vi, vii, viii } { \__enumext_tmp:n {#1} }
                                (End of definition for \l__enumext_counter_i_tl and others.)
                               Internal variables used by ref key (§13.13).
\c__enumext_counter_style_tl
 \l__enumext_ref_key_arg_tl
                                 50 \tl_const:Nn \c__enumext_counter_style_tl
                                51 { { arabic } { roman } { Roman } { alph } { Alph } }
\l__enumext_ref_the_count_tl
                                _{52} \tl_new:N \l__enumext_ref_key_arg_tl
\l__enumext_the_counter_X_tl
                                53 \tl_new:N \l__enumext_ref_the_count_tl
     \l__enumext_renew_the_count_X_tl
                                 54 \cs_set_protected:Npn \__enumext_tmp:n #1
                                   {
                                 55
                                       \tl_new:c { l__enumext_renew_the_count_#1_tl }
                                       \tl_new:c { l__enumext_the_counter_#1_tl }
                                       \tl_set:ce { l__enumext_the_counter_#1_tl } { \exp_not:c { theenumX#1 } }
                                     }
                                 60 \clist_map_inline:nn { i, ii, iii, iv, v, vi, vii, viii } { \__enumext_tmp:n {#1} }
                               (End of definition for \c__enumext_counter_style_tl and others.)
      \g__enumext_resume_int Internal variables used by resume, resume* and series keys (§13.24).
  \g__enumext_resume_vii_int
                                _{61} \int_new:N \g__enumext_resume_int
  \l enumext resume name tl
                                62 \int_new:N \g__enumext_resume_vii_int
                                63 \tl_new:N \l__enumext_resume_name_tl
      \l enumext resume active bool
                                64 \bool_new:N \l__enumext_resume_active_bool
       \g__enumext_starred_series_tl
                                               \g__enumext_standar_series_tl
                                65 \tl_new:N
       \g__enumext_standar_series_tl
                                 66 \tl_new:N
                                              \g__enumext_starred_series_tl
                                (End of definition for \g_{\text{enumext\_resume\_int}} and others.)
                               The variable \l_enumext_current_widest_dim stores the current label width, the variable \g_-
       \l__enumext_current_widest_dim
                                enumext_counter_styles_tl stores the default \langle label\ style \rangle and the variable \g_enumext_widest_-
       \g__enumext_counter_styles_tl
 \g__enumext_widest_label_tl
                                label_tl the label width. These variables are used by widest (§13.14) and label (§13.12) keys.
       \l__enumext_label_width_by_box
                                 67 \dim_new:N \l__enumext_current_widest_dim
                                 68 \tl_new:N \g__enumext_counter_styles_tl
                                 69 \tl_new:N \g__enumext_widest_label_tl
                                 70 \box_new:N \l__enumext_label_width_by_box
```

©2024 by Pablo González L

28 / 156

($End\ of\ definition\ for\ \ l_enumext_current_widest_dim\ and\ others.$)

```
\l_enumext_leftmargin_tmp_X_bool
\l_enumext_leftmargin_tmp_X_dim
\l__enumext_leftmargin_X_dim
\l__enumext_itemindent_X_dim
```

```
71 \cs_set_protected:Npn \__enumext_tmp:n #1
72 {
73    \bool_new:c { l__enumext_leftmargin_tmp_#1_bool }
74    \dim_new:c { l__enumext_leftmargin_tmp_#1_dim }
75    \dim_new:c { l__enumext_leftmargin_#1_dim }
76    \dim_new:c { l__enumext_itemindent_#1_dim }
77  }
78 \clist_map_inline:nn { i, ii, iii, iv, v, vi, vii, viii } { \__enumext_tmp:n {#1} }
```

(End of definition for \l_{-} enumext_leftmargin_tmp_X_bool and others.)

\l_enumext_multicols_above_X_skip \l_enumext_multicols_below_X_skip \g_enumext_multicols_right_X_skip \l_enumext_align_label_pos_X_str

Internal variables used by columns key (§13.21) and align key (§13.12).

(End of definition for $\l_enumext_multicols_above_X_skip$ and others.)

\g__enumext_minipage_stat_int
\l__enumext_minipage_temp_skip
\l__enumext_minipage_left_skip
\l__enumext_minipage_right_skip
\l__enumext_minipage_after_skip
\g__enumext_minipage_right_skip
\g__enumext_minipage_after_skip
\l__enumext_minipage_after_skip
\l__enumext_minipage_left_X_dim
\l__enumext_minipage_active_X_bool

Internal variables used by \miniright command (\§13.22.4) and the keys mini-right, mini-right*, minienv and mini-sep (\§13.20, \§13.22).

```
%7 \int_new:N \g__enumext_minipage_stat_int

%8 \skip_new:N \l__enumext_minipage_temp_skip

%9 \skip_new:N \l__enumext_minipage_left_skip

90 \skip_new:N \l__enumext_minipage_after_skip

91 \skip_new:N \g__enumext_minipage_after_skip

92 \skip_new:N \g__enumext_minipage_after_skip

93 \skip_new:N \g__enumext_minipage_after_skip

94 \cs_set_protected:Npn \__enumext_tmp:n #1

95 {

96   \dim_new:c { l__enumext_minipage_left_#1_dim }

97   \bool_new:c { l__enumext_minipage_active_#1_bool }

98 }

99 \clist_map_inline:nn { i, ii, iii, iv, v, vii, viii } { \__enumext_tmp:n {#1} }
```

 $(\textit{End of definition for } \ \ \, \\ \ \ \, \\ \ \, \text{End of definition for } \ \ \, \\ \ \ \, \text{g_enumext_minipage_stat_int} \ \ \, \text{and others.})$

\l_enumext_wrap_label_X_bool
\l_enumext_wrap_label_opt_X_bool
\l_enumext_start_X_int
\l_enumext_fake_item_indent_X_tl
\l_enumext_label_fill_left_X_tl
\l_enumext_label_fill_right_X_tl
\l_enumext_vspace_a_star_X_bool
\l_enumext_vspace_b_star_X_bool

The bool vars \l__enumext_wrap_label_X_bool and \l__enumext_wrap_label_opt_X_bool are used by wrap-label and wrap-label* keys ($\S13.12$), the integer \l__enumext_start_X_int are used by the start and start* keys ($\S13.14$), the token list \l__enumext_fake_item_indent_X_tl is used by itemindent key ($\S13.17.1$), the variables \l__enumext_label_fill_left_X_tl and \l_enumext_label_fill_left_X_tl are used by the align key ($\S13.12$). The boolean vars \l_enumext_vspace_-a_star_X_bool, \l_enumext_vspace_b_star_X_bool are used by above, above*, below and below* keys ($\S13.19$).

```
\cs_set_protected:Npn \__enumext_tmp:n #1
101
      \bool_new:c { l__enumext_wrap_label_#1_bool
102
      \bool_new:c { l__enumext_wrap_label_opt_#1_bool }
103
      \int_new:c { l__enumext_start_#1_int
                  { l__enumext_fake_item_indent_#1_tl }
      \tl_new:c
                  { l__enumext_label_fill_left_#1_tl
      \tl_new:c
      \tl_new:c
                  { l__enumext_label_fill_right_#1_tl }
      \bool_new:c { l__enumext_vspace_a_star_#1_bool }
      \bool_new:c { l__enumext_vspace_b_star_#1_bool }
im \clist_map_inline:nn { i, ii, iii, iv, v, vii, viii } { \__enumext_tmp:n {#1} }
```

 $(\textit{End of definition for $\setminus l_enumext_wrap_label_X_bool and others.})$

©2024 by Pablo González L 29/156

\l__enumext_anspic_body_htdp_dim

```
The variable \l__enumext_store_active_bool setting by save-ans key (§13.25.1) activates all the mech-
       \l__enumext_store_active_bool
   \l__enumext_store_name_tl
                                 anism related to \anskey, anskey*, keyans, keyans* and keyanspic environments.
   \g__enumext_store_name_tl
                                 The variable \l__enumext_store_name_tl saves the \{\langle store \ name \rangle\} set by the save-ans key of the sequence
      \l__enumext_store_anskey_arg_tl
                                 and prop list in which we will store, the variable \g__enumext_store_name_tl it's just a global copy of
      \l__enumext_store_anskey_env_tl
                                 \{\langle store\ name \rangle\} used by different functions.
      \verb|\lower| \verb| l__enumext_store_anskey_opt_tl|
                                 The variable \lower = 1.2 \text{ Losse} and the variables are the argument of \lower = 1.2 \text{ Losse} and the variables
    \l__enumext_store_current_label_tl
                                 \l__enumext_store_anskey_env_tl and \l__enumext_store_anskey_opt_tl save the \langle body \rangle and the
  \l__enumext_store_current_opt_arg_tl
                                 \langle keys \rangle of the environment anskey* (§13.30).
 \l__enumext_store_current_label_tmp_tl
                                 The variables \l__enumext_store_current_label_tl and \l__enumext_store_current_opt_arg_-
                                 tl save the current label and optional argument of \item* (§13.36) and \anspic* (§13.41.2) for the keyans,
                                 keyans* and keyanspic environments.
                                 The variable \l__enumext_store_current_label_tmp_tl is a temporary variable used by keyans,
                                 keyans* and keyanspic at various points.
                                  \bool_new:N \l__enumext_store_active_bool
                                  113 \tl_new:N \l__enumext_store_name_tl
                                                 \g__enumext_store_name_tl
                                  114 \tl_new:N
                                  115 \tl_new:N
                                                 \l__enumext_store_anskey_arg_tl
                                                 \l__enumext_store_anskey_env_tl
                                  116 \tl_new:N
                                  117 \tl_new:N
                                                 \l__enumext_store_anskey_opt_tl
                                  118 \tl_new:N
                                                  \l__enumext_store_current_label_tl
                                  119 \tl_new:N
                                                  \l__enumext_store_current_opt_arg_tl
                                  120 \tl_new:N
                                                  \l__enumext_store_current_label_tmp_tl
                                 (End of definition for \l_enumext\_store\_active\_bool and others.)
                                 Internal variables used by the command \setenumext (§13.47).
  \l__enumext_setkey_tmpa_tl
  \l__enumext_setkey_tmpb_tl
                                  \tl_new:N \l__enumext_setkey_tmpa_tl
 \l__enumext_setkey_tmpa_int
                                  122 \tl_new:N \l__enumext_setkey_tmpb_tl
 \l__enumext_setkey_tmpa_seq
                                  123 \int_new:N \l__enumext_setkey_tmpa_int
                                  \seq_new:N \l__enumext_setkey_tmpa_seq
 \l__enumext_setkey_tmpb_seg
                                  \seq_new:N \l__enumext_setkey_tmpb_seq
                                 (End of definition for \l_enumext_setkey_tmpa_tl and others.)
                                 Internal variables used by the \printkeyans command (§13.46) and \foreachkeyans command (§13.49).
    \l__enumext_meta_path_tl
        \l__enumext_foreach_print_seq
                                 126 \tl_new:N \l__enumext_meta_path_tl
                                 \seq_new:N \l__enumext_foreach_print_seq
     \l__enumext_foreach_name_prop_tl
                                 \tl_new:N \l__enumext_foreach_name_prop_tl
   \g__enumext_foreach_default_keys_tl
                                  129 \tl_new:N \g__enumext_foreach_default_keys_tl
                                 (End of definition for \l_enumert_meta_path_tl and others.)
                                 Internal variables used by command \printkeyans (\§13.46), show-pos key (\§13.26), item-sym* key (\§13.34),
   \l__enumext_print_keyans_starred_tl
                                 save-key key (§13.26.2) and "storing structure".
    \l__enumext_print_keyans_star_bool
       \l__enumext_mark_position_str
                                                \l__enumext_print_keyans_starred_tl
                                  130 \tl_new:N
       \g__enumext_item_symbol_aux_tl
                                  \bool_new:N \l__enumext_print_keyans_star_bool
                                  \str_new:N \l__enumext_mark_position_str
       \l__enumext_print_keyans_X_tl
                                  133 \tl_new:N \g__enumext_item_symbol_aux_tl
      \l__enumext_store_save_key_X_tl
                                  \cs_set_protected:Npn \__enumext_tmp:n #1
    \l__enumext_store_save_key_X_bool
                                  135
  \l__enumext_store_upper_level_X_bool
                                         \tl_new:c { l__enumext_print_keyans_#1_tl
                                  136
                                         \tl_new:c { l__enumext_store_save_key_#1_tl
                                  137
                                                                                                  }
                                         \bool_new:c { l__enumext_store_save_key_#1_bool
                                         \bool_new:c { l__enumext_store_upper_level_#1_bool }
                                  (End of definition for \l_enumext\_print\_keyans\_starred\_tl and others.)
                                 Internal variables used by keyanspic environment and \anspic command (§13.41.1).
 \l__enumext_anspic_args_seq
    \l__enumext_anspic_mini_width_dim
                                  \seq_new:N \l__enumext_anspic_args_seq
\l enumext anspic above int
                                  143 \dim_new:N \l__enumext_anspic_mini_width_dim
                                  \int_new:N \l__enumext_anspic_above_int
\l__enumext_anspic_below_int
                                  145 \int_new:N \l__enumext_anspic_below_int
   \l__enumext_anspic_label_above_bool
                                  146 \bool_new:N \l__enumext_anspic_label_above_bool
      \l__enumext_anspic_mini_pos_str
                                  \str_new:N \l__enumext_anspic_mini_pos_str
    \g__enumext_keyans_pic_parsep_skip
                                  \skip_new:N \g__enumext_keyans_pic_parsep_skip
\l__enumext_anspic_label_box
                                  149 \box_new:N \l__enumext_anspic_label_box
\l__enumext_anspic_body_box
                                  150 \box_new:N \l__enumext_anspic_body_box
    \l__enumext_anspic_label_htdp_dim
                                  {}_{^{151}}\ \backslash \text{dim\_new:N} \ \backslash \text{l\_\_enumext\_anspic\_label\_htdp\_dim}
```

 $\label{eq:local_local_local_local} $$ \dim_new:N $$ \l_enumext_anspic_body_htdp_dim $$$

©2024 by Pablo González L

30 / 156

```
(End of definition for \l_enumext_anspic_args_seq and others.)
```

```
Internal variables used by "internal check answer" mechanism (§13.25.3) used by the check-ans and no-
       \l__enumext_check_answers_bool
                               store keys and check for starred commands \item* in keyans and keyans* environments and \anspic*
       \g__enumext_check_ans_key_bool
                               in keyanspic environment.
   \l__enumext_check_start_line_env_tl
    \g__enumext_check_starred_cmd_int
                                _{^{153}} \bool_new:N \l__enumext_check_answers_bool
 \g__enumext_item_anskey_int
                                _{^{154}} \bool_new:N \g__enumext_check_ans_key_bool
                                155 \tl_new:N \l__enumext_check_start_line_env_tl
 \g__enumext_item_number_int
                                _{\mbox{\scriptsize 156}} \int_new:N \g__enumext_check_starred_cmd_int
\g__enumext_item_number_bool
                                'int_new:N \g__enumext_item_anskey_int
     \g__enumext_item_answer_diff_int
                                158 \int_new:N \g__enumext_item_number_int
                                159 \bool_new:N \l__enumext_item_number_bool
                                _{160} \int_new:N \g__enumext_item_answer_diff_int
                               (\textit{End of definition for} \setminus \texttt{l}\_\texttt{enumext\_check\_answers\_bool} \ \ \textit{and others.})
   \l__enumext_hyperref_bool
                               The boolean variable \l_enumext_hyperref_bool will determine if the hyperref package is present or
       \l__enumext_footnotes_key_bool
                               load in memory (§13.8). The boolean variable \l__enumext_footnotes_key_bool determine if hyperref
                               is load with key hyperfootnotes=true.
                                \text{\lool_new:N \l__enumext_hyperref_bool}
                                162 \bool_new:N \l__enumext_footnotes_key_bool
                               \l__enumext_newlabel_arg_one_tl
                               Internal variables used by save-ref key (§13.26). The variables \l__enumext_label_copy_X_tl corre-
                               spond to temporary copies of the (labels) defined by level on which operations will be performed.
      \l__enumext_newlabel_arg_two_tl
       \l__enumext_write_aux_file_tl
                               \l__enumext_label_copy_X_tl
                               used to form the arguments passed to the function \__enumext_newlabel:nn (§13.8) and the variable
                                \l__enumext_write_aux_file_tl will be in charge of executing the writing code in the .aux file.
                                163 \tl_new:N \l__enumext_newlabel_arg_one_tl
                                _{164} \tl_new:N \l__enumext_newlabel_arg_two_tl
                                165 \tl_new:N \l__enumext_write_aux_file_tl
                                166 \cs_set_protected:Npn \__enumext_tmp:n #1
                                       \tl_new:c { l__enumext_label_copy_#1_tl }
                                _{170} \clist_map_inline:nn { i, ii, iii, iv, v, vi, vii, viii } { \__enumext_tmp:n {#1} }
                               (End\ of\ definition\ for\ \l_enumext_newlabel\_arg\_one\_tl\ and\ others.)
                               Internal variables used for redefinition of \footnote (§13.42.4).
    \g__enumext_footnote_int
\g__enumext_footnote_arg_seq
                                ''71 \int_new:N \g__enumext_footnote_int
\g__enumext_footnote_int_seq
                                \seq_new:N \g__enumext_footnote_arg_seq
                                \seq_new:N \g__enumext_footnote_int_seq
                               Internal variables used by enumext* and keyans* environments.
      \l enumext item starred X bool
     l__enumext_item_column_pos_X_int
                                \cs_set_protected:Npn \__enumext_tmp:n #1
     \g__enumext_item_count_all_X_int
                                175
       \l__enumext_joined_item_X_int
                                       \bool_new:c { l__enumext_item_starred_#1_bool
                                176
                                       \int_new:c { l__enumext_item_column_pos_#1_int }
    \l enumext joined item aux X int
                                       \int_new:c { g__enumext_item_count_all_#1_int
                                178
      \l__enumext_tmpa_X_int
                                       \int_new:c { l__enumext_joined_item_#1_int
                                179
      \l__enumext_tmpa_X_dim
                                       \int_new:c { l__enumext_joined_item_aux_#1_int }
 \l__enumext_item_text_X_box
                                       \int_new:c { l__enumext_tmpa_#1_int
                                                                                           }
      \l__enumext_joined_width_X_dim
                                       \label{local_dim_new} $$\dim_{new:c} \{ l_{enumext_tmpa_\#1_dim} \}$$
                                182
                                                                                           }
\l__enumext_item_width_X_dim
                                183
                                       \box_new:c { l__enumext_item_text_#1_box
                                                                                           }
     \g__enumext_item_symbol_aux_X_tl
                                       \dim_new:c { l__enumext_joined_width_#1_dim
                                                                                           }
                                184
       \l__enumext_align_label_X_str
                                       \dim_new:c { l__enumext_item_width_#1_dim
                                                                                           }
                                185
   \g__enumext_minipage_active_X_bool
                                                   { g__enumext_item_symbol_aux_#1_tl
                                       \tl_new:c
                                186
     \l__enumext_miniright_code_X_box
                                       \str_new:c { l__enumext_align_label_#1_str
                                187
    \g__enumext_minipage_center_X_bool
                                       \bool_new:c { g__enumext_minipage_active_#1_bool }
                                       \box_new:c { l__enumext_miniright_code_#1_box
     \g__enumext_minipage_right_X_dim
                                       \bool_new:c { g__enumext_minipage_center_#1_bool }
    \g__enumext_minipage_right_X_skip
                                       \dim_new:c { g__enumext_minipage_right_#1_dim
                                       \skip_new:c { g__enumext_minipage_right_#1_skip
                                \clist_map_inline:nn { vii, viii } { \__enumext_tmp:n {#1} }
```

©2024 by Pablo González L 31/156

($End\ of\ definition\ for\ \ l_enumext_item_starred_X_bool\ and\ others.$)

\c__enumext_all_envs_clist

An internal clist-var variable to run with __enumext_tmp:n.

```
195 \clist_const:Nn \c__enumext_all_envs_clist
      {level-1}{i}, {level-2}{ii}, {level-3}{iii}, {level-4}{iv},
      {keyans}{v}, {enumext*}{vii}, {keyans*}{viii}
```

(End of definition for $\c_enumert_all_envs_clist$.)

13.5 Some utility functions

\seq_use:NV

\keys_precompile:neN Non-standard kernel variants used by the \printkeyans command (§13.46) and \foreachkeyans command (§13.49).

```
200 \cs_generate_variant:Nn \keys_precompile:nnN { neN }
201 \cs_generate_variant:Nn \seq_use:Nn { NV }
```

(End of definition for \keys_precompile:neN and \seq_use:NV.)

__enumext_at_begin_document:n

A internal "hook" function used for copying plain list and minipage environments definition and hyperref detection.

```
202 \cs_new_protected:Npn \__enumext_at_begin_document:n #1
203 {
      \hook_gput_code:nnn {begindocument} {enumext} { #1 }
    7
```

(End of definition for $_=$ enumext_at_begin_document:n.)

__enumext_after_env:nn \ enumext before env:nn

A internal "hook" functions for execute code mini-right and mini-right* keys outside the enumext* and keyans* environments and print check-ans outside the enumext and enumext* environments.

```
206 \cs_new_protected:Npn \__enumext_after_env:nn #1 #2
      \hook_gput_code:nnn {env/#1/after} {enumext} {#2}
   }
\cs_new_protected:Npn \__enumext_before_env:nn #1 #2
211
      \hook_gput_code:nnn {env/#1/before} {enumext} {#2}
```

(End of definition for $\ensuremath{\text{c-enumext_after_env:nn}}$ and $\ensuremath{\text{c-enumext_before_env:nn.}}$)

_enumext_level: Function for check current level in enumext.

```
214 \cs_new:Nn \__enumext_level:
      \int_to_roman:n { \l__enumext_level_int }
```

(End of definition for __enumext_level:.)

__enumext_if_is_int:nF __enumext_if_is_int:nTF

__enumext_if_is_int:nT A conditional function to know if the variable we are passing is an integer used by start and widest keys. This function is taken directly from the answer given by Henri Menke in How to test if an expl3 function argument is an integer expression?.

```
218 \prg_new_protected_conditional:Npnn \__enumext_if_is_int:n #1 { T, F, TF }
219
      \regex_match:nnTF { ^[\+\-]?[\d]+$ } {#1} % $
        { \prg_return_true: }
         { \prg_return_false: }
```

 $(End\ of\ definition\ for\ __enumext_if_is_int:nT,\ __enumext_if_is_int:nF.)$

__enumext_regex_counter_style:

The internal function __enumext_regex_counter_style: replace the '*' with the actual counter of the running level and is used by the ref key. It loops through the defined counter styles in \c_enumext_counter_style_tl and replace '*' by real command, for example, looking for \arabic* and replacing that by $\arabic{\langle counter \rangle}$ defined on the current level.

```
\cs_new_protected:Nn \__enumext_regex_counter_style:
225
       \tl_map_inline:Nn \c__enumext_counter_style_tl
227
           \regex_replace_once:nnN { \c{##1}\* }
             { \c{##1}\cB{\u{l_enumext_ref_the_count_tl}\cE} } \l_enumext_ref_key_arg_tl
    7
©2024 by Pablo González L
```

(End of definition for __enumext_regex_counter_style:.)

__enumext_show_length:nnn

Internal function used by show-length key to show "all lengths" calculated and use in enumext, enumext*, keyans and keyans* environments.

(End of definition for $_=$ enumext_show_length:nnn.)

__enumext_unskip_unkern:

The function __enumext_unskip_unkern: will remove the last $\langle skip \rangle$ or $\langle kern \rangle$ at execution time using the values 11 and 12 of \lastnodetype to apply \unskip or \unkern according to the case.

13.5.1 Utilities for environments and levels

__enumext_is_not_nested:
 __enumext_is_on_first_level:

The function $_$ enumext_is_not_nested: set the variables $_$ enumext_standar_bool and $_$ enumext_starred_bool to "true" only if the environments enumext and enumext* are nested in each other and save the environment name in $_$ enumext_envir_name_tl.

```
254 \cs_new_protected:Nn \__enumext_is_not_nested:
    {
255
      \str_case:en { \@currenvir }
256
        {
257
           {enumext}
258
               \tl_set:Nn \l__enumext_envir_name_tl { enumext }
               \bool_lazy_and:nnT
                 { \bool_not_p:n { \g__enumext_standar_bool } }
                 { \int_compare_p:nNn { \l__enumext_level_h_int } = { 0 } }
                 {
                   \bool_gset_true:N \g__enumext_standar_bool
                }
            }
           {enumext*}
            {
               \tl_set:Nn \l__enumext_envir_name_tl { enumext* }
               \bool_lazy_and:nnT
                 { \bool_not_p:n { \g__enumext_starred_bool } }
                 { \int_compare_p:nNn { \l__enumext_level_int } = { 0 } }
                   \bool_gset_true:N \g__enumext_starred_bool
                 }
            }
        }
278
```

The function __enumext_is_on_first_level: will set the variables \l__enumext_standar_first_bool ($\S13.25.1$), \l__enumext_starred_first_bool ($\S13.25.1$) and \l__enumext_anskey_env_bool ($\S13.30$) to "true" only if the environment is not nested and we are in the "first level" of it . We will also save the start line number of each environment in the variable \g__enumext_start_line_tl and the name of each environment in the variable \g__enumext_envir_name_tl to use in messages related to the checkans key and .log file.

```
280 \cs_new_protected:Nn \__enumext_is_on_first_level:
281
       \bool_lazy_all:nT
282
283
         {
           { \bool_if_p:N \g__enumext_standar_bool }
284
           { \int_compare_p:nNn { \l__enumext_level_int } = { 1 } }
           { \int_compare_p:nNn { \l__enumext_level_h_int } = { 0 } }
         }
         {
           \bool_set_true:N \l__enumext_standar_first_bool
           \bool_set_true:N \l__enumext_anskey_env_bool
           \tl_gset:Nn \g__enumext_envir_name_tl { enumext }
           \tl_gset:Ne \g__enumext_start_line_tl
             {
               on ~ line ~ \exp_not:V \inputlineno
295
         }
       \bool_lazy_all:nT
297
           { \bool_if_p:N \g__enumext_starred_bool }
           { \int_compare_p:nNn { \l__enumext_level_h_int } = { 1 } }
           { \int_compare_p:nNn { \l__enumext_level_int } = { 0 } }
         }
         {
           \bool_set_true:N \l__enumext_starred_first_bool
           \bool_set_true:N \l__enumext_anskey_env_bool
           \tl_gset:Nn \g__enumext_envir_name_tl { enumext* }
           \tl_gset:Ne \g__enumext_start_line_tl
               on ~ line ~ \exp_not:V \inputlineno
         }
     }
(End of definition for \__enumext_is_not_nested: and \__enumext_is_on_first_level:.)
```

__enumext_keyans_name_and_start:

The function __enumext_keyans_name_and_start: will save the start line number and name of the environments keyans, keyans* and keyanspic in the variables \l__enumext_check_start_line_env_tl and \l__enumext_envir_name_tl to use in the __enumext_check_starred_cmd:n function.

```
313 \cs_new_protected:Nn \__enumext_keyans_name_and_start:
      \str_case:en { \@currenvir }
316
          {keyans}
              \tl_set:Nn \l__enumext_envir_name_tl { keyans }
              \tl_set:Ne \l__enumext_check_start_line_env_tl
                   in ~ 'keyans' ~ start ~ on ~ line ~ \exp_not:V \inputlineno
                }
323
            }
324
          {keyans*}
               \tl_set:Nn \l__enumext_envir_name_tl { keyans* }
               \tl_set:Ne \l__enumext_check_start_line_env_tl
                   in ~ 'keyans*' ~ start ~ on ~ line ~ \exp_not:V \inputlineno
                 }
            }
          {keyanspic}
               \tl_set:Nn \l__enumext_envir_name_tl { keyanspic }
               \tl_set:Ne \l__enumext_check_start_line_env_tl
                   in ~ 'keyanspic' ~ start ~ on ~ line ~ \exp_not:V \inputlineno
                 }
            }
        }
    }
```

 $(\textit{End of definition for } \verb|_enumext_keyans_name_and_start:.)$

©2024 by Pablo González L 34/156

13.5.2 Utilities for log and terminal

 The function __enumext_reset_global_vars: will be passed to the function __enumext_execute_-after_env: and will return the global variables to their default values after being used.

```
343 \cs_new_protected:Nn \__enumext_reset_global_vars:
   {
344
       \__enumext_reset_global_int:
345
       \__enumext_reset_global_bool:
346
       \__enumext_reset_global_tl:
347
348
349 \cs_new_protected:Nn \__enumext_reset_global_int:
350
       \int_gzero:N \g__enumext_item_number_int
       \verb|\int_gzero:N \  \  \| g_{\_}enumext_item_anskey_int|
       \int_gzero:N \g__enumext_item_answer_diff_int
    }
\cs_new_protected:Nn \__enumext_reset_global_bool:
356
       \bool_gset_false:N \g__enumext_check_ans_key_bool
357
       \bool_gset_false:N \g__enumext_standar_bool
358
      \bool_gset_false:N \g__enumext_starred_bool
359
    }
361 \cs_new_protected:Nn \__enumext_reset_global_tl:
362
       \tl_gclear:N \g__enumext_store_name_tl
363
      \tl_gclear:N \g__enumext_start_line_tl
       \tl_gclear:N \g__enumext_envir_name_tl
```

(End of definition for $_$ enumext_reset_global_vars: and others.)

__enumext_log_global_vars:
__enumext_log_answer_vars:

The function __enumext_log_global_vars: will be passed to the function __enumext_execute_-after_env: and write to the .log file the number of elements saved in the *prop list* and *sequence* created by the save-ans key along with the value of the integer variable created for the resume key.

The function __enumext_log_answer_vars: will be passed to the function __enumext_execute_-after_env: and write to the .log file the number of items and answers along with the difference between them

 $(\textit{End of definition for } \c\c enumert_log_global_vars: and \c\c enumert_log_answer_vars:.)$

13.6 Copying list and minipage environments

The list environment provided by LETEX has the following plain form:

```
\label{eq:cone} $$ \left( arg \ one \right) \left\{ \left\langle arg \ two \right\rangle \right\} $$ \left( opt \right) $$ endlist
```

And minipage environment provided by LTFX has the following (simplified) plain form:

```
\begin{tabular}{ll} $$\min[page[\langle pos \rangle][\langle height \rangle][\langle inner-pos \rangle]\{\langle width \rangle\} \\ & \langle internal\ implement \rangle \\ \begin{tabular}{ll} \end{tabular} $$\end{tabular}
```

As a precaution we copy them using __enumext_at_begin_document:n in case any package redefines the list environment or a related command.

◆ For compatibility with tagged PDF we should use \NewCommandCopy and not \cs_new_eq:NN for \item. When tagged PDF is active \item is redefined using ltcmd (see latex-lab-block).

```
©2024 by Pablo González L
```

```
\__enumext_start_list:nn
  \__enumext_stop_list:
  \__enumext_item_std:w
  \__enumext_minipage:w
  \__enumext_endminipage:
```

The functions __enumext_start_list:nn and __enumext_stop_list: correspond to copies of \list and \endlist from plain definition of list, the function __enumext_item_std:w is a copy of the \item command.

The functions __enumext_minipage:wand __enumext_endminipage: correspond to copies of \minipage and \endminipage from plain definition of minipage environment.

(End of definition for $_$ enumext_start_list:nn and others.)

13.7 The internal minipage environment

__enumext_internal_mini_page:
 __enumext_mini_env*

The function __enumext_internal_mini_page: creates a internal __enumext_mini_page environment (custom version of minipage) setting the \if@minipage switch to "false" to allow spaces at the "above" of the environment, plus we will add \skip_vertical:N \c_zero_skip to maintain alignment on "top" in the first part and \skip_vertical:N \c_zero_skip in the second part to allow spaces "below". This environment will be used internally by the mini-env key, it is not documented in the user interface and is for internal use only. This function is passed to the function __enumext_safe_exec: in the enumext environment definition (§13.38) and __enumext_safe_exec_vii: in the enumext* environment definition (§13.43)

```
393 \cs_new_protected:Nn \__enumext_internal_mini_page:
    {
394
      \int_compare:nNnT { \l__enumext_level_int } = { 0 }
395
        {
396
           \DeclareDocumentEnvironment{__enumext_mini_page}{ m }
397
             {
               \__enumext_minipage:w [ t ] { ##1 }
                 \legacy_if_gset_false:n { @minipage }
                 \skip_vertical:N \c_zero_skip
             }
             {
                 \skip_vertical:N \c_zero_skip
               \ enumext endminipage:
        }
    }
```

(End of definition for __enumext_internal_mini_page: and __enumext_mini_env*.)

13.8 Compatibility with hyperref and footnotehyper

First we define the necessary rules using "hooks" to determine if the hyperref package is loaded.

```
hook_gput_code:nnn { begindocument } { enumext } { \__enumext_after_hyperref: }
hook_gset_rule:nnnn { begindocument } { enumext } { after } { hyperref }
```

__enumext_after_hyperref:
__enumext_hypertarget:nn
__enumext_phantomsection:

The function __enumext_after_hyperref: sets the state of the boolean variable \l__enumext_-hyperref_bool to "true" if the package is loaded. At this point we will use the public macro \IfHyperBoolean to determine if the hyperfootnotes=true key is present, if so, we set the state of the boolean variable __enumext_footnotes_key_bool to "true".

```
411 \cs_new_protected:Nn \__enumext_after_hyperref:
       \IfPackageLoadedTF { hyperref }
413
414
           \msg_info:nnn { enumext } { package-load } { hyperref }
415
           \bool_set_true:N \l__enumext_hyperref_bool
416
           \IfHyperBoolean{hyperfootnotes}
417
             {
418
               % \typeout{hyperfootnotes=true}
               \bool_set_true:N \l__enumext_footnotes_key_bool
             }
             {
```

©2024 by Pablo González L

If the state of the variable \l__enumext_footnotes_key_bool is true we will check if the package footnotehyper is loaded, in case it is not present, we will set the value of \l__enumext_footnotes_key_bool to false and we will redefine \footnote.

```
\text{bool_if:NT \l_enumext_footnotes_key_bool}

{

\text{IfPackageLoadedTF { footnotehyper }}

\text{displayed by the section of the s
```

The functions __enumext_hypertarget:nn and __enumext_phantomsection: correspond to the internal copies of \hypertarget and \phantomsection. If the boolean variable \l__enumext_hyperref_bool is false the functions __enumext_hypertarget:nn and __enumext_phantomsection: will be disabled.

```
\dashbool_if:NTF \l__enumext_hyperref_bool

\dashbool_if:NTF \l__enumext_hyperref_bool

\dashbool_if:NTF \l__enumext_hyperref_bool

\dashbool_if:NTF \l__enumext_hyperref_bool

\dashbool_if:NTF \l__enumext_hypertarget:nn \hypertarget

\dashbool_if:NTF \l_enumext_hypertarget:nn \hypertarget

\dashbool_if:NTF \l_enum
```

 $(\textit{End of definition for } \verb|_= enumext_after_hyperref:, \verb|_= enumext_hypertarget:nn|, and \verb|_== enumext_phantomsection:.)|$

__enumext_newlabel:nn

The function __enumext_newlabel:nn write the information to the .aux file when using the save-ref key. The arguments taken by the function are:

```
#1: \l_enumext_newlabel_arg_one_tl
#2: \l_enumext_newlabel_arg_two_tl
```

The trick here is to manage the number of arguments passed to \newlabel{#1}{#2} according to the presence of the hyperref package.

```
449 \cs_new_protected:Npn \__enumext_newlabel:nn #1 #2
450
451
      \protected@write \@auxout { }
452
           \token_to_str:N \newlabel {#1}
             {
               {#2}
               \bool_if:NT \l__enumext_hyperref_bool
                 { { \thepage } {#1} }
               { }
458
             }
        }
       \__enumext_hypertarget:nn {#1} { }
       \__enumext_phantomsection:
    }
```

(End of definition for $_$ enumext_newlabel:nn.)

13.9 Definition of public dimension

The package enumext only provides a single public dimension \itemwidth and is intended for user convenience only and is not for internal use as such. This dimension is set in all environments and is only used by the wrap-ans key at its default value.

```
464 \dim_zero_new:N \itemwidth
```

13.10 Definition of counters

__enumext_define_counters:Nn __enumext_define_counters:cn To create the necessary "counters" we must first make sure that they are not already defined by the user or a package such as enumitem, otherwise a error will be returned and the package loading will be aborted. The arguments taken by the function are:

#1: A token list \l__enumext_counter_X_tl for "store" the counter's name.

#2: The counter's name.

(End of definition for $_$ enumext_define_counters:Nn.)

enumXii The counters created here are enumXi, enumXii, enumXiii and enumXiv for enumext environment, enumXv for keyans environment, enumXvii for keyanspic environment, enumXviii for the keyans* environments.

```
enumXiv
            474 \__enumext_define_counters:Nn \l__enumext_counter_i_tl
                                                                                √ enumXi
                                                                                              }
   enumXv
            475 \__enumext_define_counters:Nn \l__enumext_counter_ii_tl
                                                                                √ enumXii
  enumXvi
            \label{eq:local_local_local_local_local_local} $$ \_\ensuremath{\text{enum}}$ = \counter_iii_tl $$ enumXiii $$
            _{\mbox{\scriptsize 477}} \__enumext_define_counters:Nn \l__enumext_counter_iv_tl ~\{ enumXiv
 enumXvii
            478 \__enumext_define_counters:Nn \l__enumext_counter_v_tl
                                                                                { enumXv
enumXviii
            479 \__enumext_define_counters:Nn \l__enumext_counter_vi_tl
                                                                                { enumXvi
            480 \__enumext_define_counters:Nn \l__enumext_counter_vii_tl
                                                                                √ enumXvii
            481 \__enumext_define_counters:Nn \l__enumext_counter_viii_tl { enumXviii
```

(End of definition for enumXi and others.)

13.11 Definition of labels

This part of the code is inspired by the enumitem package. The idea is to be able to access the counters using \arabic*, \Alph*, \alph*, \Roman* and \roman* to use them in the label key.

__enumext_register_counter_style:Nn

These $\langle counters \rangle$ will be used as default $\langle labels \rangle$ if the label key is not used for the different levels of the enumext, enumext*, keyans and keyans* environments, so it is necessary to get a default value for labelwidth from these $\langle labels \rangle$ at the same time.

```
482 \cs_new_protected:Npn \__enumext_register_counter_style:Nn #1 #2
483 {
484     \tl_const:cn { c__enumext_widest_ \cs_to_str:N #1 _tl } {#2}
485     \tl_gput_right:Nn \g__enumext_counter_styles_tl {#1}
486 }
487 \__enumext_register_counter_style:Nn \arabic { 0 }
488 \__enumext_register_counter_style:Nn \Alph { M }
489 \__enumext_register_counter_style:Nn \alph { m }
490 \__enumext_register_counter_style:Nn \Roman { VIII }
491 \__enumext_register_counter_style:Nn \roman { viii }
```

 $(\textit{End of definition for } \c\c enumert_register_counter_style:Nn.)$

__enumext_label_width_by_box:Nn __enumext_label_width_by_box:cv The function $\ensuremath{\verb|_enumext_label_width|}$ by $\ensuremath{\verb|by_box:Nn|}$ set the default $\ensuremath{\verb|labelwidth|}$ using a box width if no labelwidth key is passed.

```
d92 \cs_new_protected:Npn \__enumext_label_width_by_box:Nn #1 #2
d93 {
d94    \hbox_set:Nn \l__enumext_label_width_by_box {#2}
d95    \dim_set:Nn #1 { \box_wd:N \l__enumext_label_width_by_box }
d96 }
d97 \cs_generate_variant:Nn \__enumext_label_width_by_box:Nn { cv }
```

 $(\textit{End of definition for } \c enumert_label_width_by_box:Nn.)$

__enumext_label_style:Nnn
__enumext_label_style:cvn

The function __enumext_label_style: Nnn is used by the label key to creates the variables containing the $\langle label\ style \rangle$ and will allow to use \arabic*, \Alph*, \alph*, \Roman* and \roman* as arguments. It loops through the defined counter styles in \g__enumext_counter_styles_tl (\arabic, \alph, \Alph, \roman, and \Roman) for example, looking for \roman* and replacing that by \roman{\cunter\}, and doing the same for the \g_enumext_widest_label_tl to keep both in sync.

```
498 \cs_new_protected:Npn \__enumext_label_style:Nnn #1 #2 #3
©2024 by Pablo González L
```

```
\tl_clear_new:N #1
      \tl_put_right:Ne #1 { \tl_trim_spaces:n {#3} }
      \tl_gset_eq:NN \g__enumext_widest_label_tl #1
      \tl_map_inline:Nn \g__enumext_counter_styles_tl
        {
504
          \tl_replace_all:Nne #1 { ##1* } { \exp_not:N ##1 {#2} }
          \tl_greplace_all:Nne \g__enumext_widest_label_tl { ##1* }
             { \tl_use:c { c_enumext_widest_ \cs_to_str:N ##1 _tl } }
        }
      \__enumext_label_width_by_box:Nn \l__enumext_current_widest_dim
        { \tl_use:N \g__enumext_widest_label_tl }
511
      \tl_set_eq:cN { the #2 } #1
513 \cs_generate_variant:Nn \__enumext_label_style:Nnn { cvn }
```

(End of definition for $_$ enumext_label_style:Nnn.)

13.12 Setting keys associated with label

When tagged PDF is active \makelabel is redefined using \makebox to work correctly (§13.33). From the user side it is convenient to have a key that allows using this redefinition with \makebox without having \IfDocumentMetadataTF active.

We define the key mode-box only for the "first level" of enumext and enumext* environments. mode-box

```
514 \cs_set_protected:Npn \__enumext_tmp:n #1
      \keys_define:nn { enumext / #1 }
        {
          mode-box .bool_set:N = \l__enumext_mode_box_bool,
          mode-box .initial:n = false,
          mode-box .value_forbidden:n = true,
520
521
522
523 \clist_map_inline:nn { level-1, enumext* } { \__enumext_tmp:n {#1} }
```

(End of definition for mode-box.)

font Definition of keys font, labelsep, labelwidth, wrap-label and wrap-label* keys for enumext and keyans environments.

```
labelwidth
 wrap-label
wrap-label*
```

labelsep

```
\cs_set_protected:Npn \__enumext_tmp:nn #1 #2
525
   {
      \keys_define:nn { enumext / #1 }
526
527
        {
          font
                      .tl_set:c = { l__enumext_label_font_style_#2_tl },
528
                      .value_required:n = true,
          font
529
          labelsep
                      .dim_set:c = { l__enumext_labelsep_#2_dim },
530
                      .initial:n = {0.3333em},
          labelsep
          labelsep
                      .value_required:n = true,
          labelwidth .dim_set:c = { l__enumext_labelwidth_#2_dim },
          labelwidth .value_required:n = true,
          wrap-label .cs_set_protected:cp = { __enumext_wrapper_label_#2:n } ##1,
          wrap-label .initial:n = {##1},
          wrap-label .value_required:n = true,
          wrap-label* .code:n = {
538
                                   \bool_set_true:c { l__enumext_wrap_label_opt_#2_bool }
                                   \keys_set:nn { enumext / #1 } { wrap-label = {##1} }
                                },
          wrap-label* .value_required:n = true,
        }
545 \clist_map_inline:Nn \c__enumext_all_envs_clist { \__enumext_tmp:nn #1 }
```

(End of definition for font and others.)

¶ In this point, the following are set __enumext_wrapper_label_X:n which will be used by __enumext_make_label: for the different levels of the enumext environment and is set to __enumext_wrapper_label_v:n which will be used by __enumext_keyans_make_label: for keyans and keyanspic environments.

align The align key is implemented differently for "starred" and "non starred" environments. For compatibility with tagged PDF we must set \l__enumext_align_label_pos_X_str.

```
_{\rm 546} \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
```

```
\keys_define:nn { enumext / #1 }
          align .choice:.
          align / left
                           .code:n =
                             {
                               \tl_clear:c { l__enumext_label_fill_left_#2_tl }
                               \tl_set:cn { l__enumext_label_fill_right_#2_tl } { \hfill }
                               \str_set:cn { l__enumext_align_label_pos_#2_str } { l }
                             },
          align / right
                           .code:n =
                             {
                               \tl_set:cn { l__enumext_label_fill_left_#2_tl } { \hfill }
                               \tl_clear:c { l__enumext_label_fill_right_#2_tl }
                               \str_set:cn { l__enumext_align_label_pos_#2_str } { r }
                             },
          align / center
                           .code:n =
563
                             {
                               \tl_set:cn { l__enumext_label_fill_left_#2_tl } { \hfill }
                               \tl_set:cn { l__enumext_label_fill_right_#2_tl } { \hfill }
                               \str_set:cn { l__enumext_align_label_pos_#2_str } { c }
                             },
          align / unknown .code:n =
                             \msg_error:nneee { enumext } { unknown-choice }
                               { align } { left, ~ right, ~ center } { \exp_not:n {##1} },
          align .initial:n = left,
          align .value_required:n = true,
576 \clist_map_inline:nn
577
      {level-1}{i}, {level-2}{ii}, {level-3}{iii}, {level-4}{iv}, {keyans}{v}
578
    }
    { \__enumext_tmp:nn #1 }
580
581 \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
582
      \keys_define:nn { enumext / #1 }
583
584
          align .choice:,
          align / left
                           .code:n = \str_set:cn { l__enumext_align_label_#2_str } { l },
          align / right
                           .code:n = \str_set:cn { l__enumext_align_label_#2_str } { r },
          align / center .code:n = \str_set:cn { l__enumext_align_label_#2_str } { c },
          align / unknown .code:n =
                             \msg_error:nneee { enumext } { unknown-choice }
                               { align } { left, ~ right, ~ center } { \exp_not:n {##1} },
          align .initial:n = left,
          align .value_required:n = true,
593
594
596 \clist_map_inline:nn { {enumext*}{vii}, {keyans*}{viii} } { \__enumext_tmp:nn #1 }
```

 $(End\ of\ definition\ for\ align.)$

13.13 Setting label and ref keys

The implementation of the keys label and ref are part of the core of the package enumext, here the default values for $\langle label \rangle$, the value of the variables \l__enumext_label_X_tl, the default values for \labelwidth and the "label and ref" system.

13.13.1 Define and set label and ref keys for enumext environment

Here we set the default $\langle labels \rangle$ of the four levels of enumext environment, along with the default value for labelwidth key and ref key.

```
ref
\l__enumext_label_i_tl
\l__enumext_label_ii_tl
\l__enumext_label_iii_tl
\l__enumext_label_iv_tl
```

```
label .initial:n = #3.
          label .value_required:n = true,
                 .code:n = \__enumext_standar_ref:n {##1},
          ref
                 .value_required:n = true,
          ref
        }
611
612
613 \__enumext_tmp:nnn { level-1 } { i } { \arabic*.}
614 \__enumext_tmp:nnn { level-2 } { ii } { (\alph*) }
615 \__enumext_tmp:nnn { level-3 } { iii } { \roman*. }
616 \__enumext_tmp:nnn { level-4 } { iv } { \Alph*. }
```

(End of definition for label and others.)

__enumext_standar_ref:n __enumext_standar_ref:

The __enumext_standar_ref:n first we will pass the key argument to \l__enumext_ref_key_arg_tl and we will analyze its state, if it is not *empty* we will make a copy of the current counter in \l__enumext _ref_the_count_tl and we will execute the function __enumext_regex_counter_style: which will return the modified \l__enumext_ref_key_arg_tl and we make the value of \l__enumext_ref_the_count_tl the same as that \l__enumext_the_counter_X_tl which contains \theenumX and finally we set \l__enumext_renew_the_count_X_tl with the renewed command.

```
617 \cs_new_protected:Npn \__enumext_standar_ref:n #1
618
       \tl_set:Nn \l__enumext_ref_key_arg_tl {#1}
      \tl_if_empty:NTF \l__enumext_ref_key_arg_tl
          \msg_error:nnn { enumext } { key-ref-empty } { enumext }
        }
        {
          \tl_set_eq:Nc
            \l__enumext_ref_the_count_tl { l__enumext_counter_ \__enumext_level: _tl }
           \__enumext_regex_counter_style:
          \tl_set_eq:Nc
            \l__enumext_ref_the_count_tl { l__enumext_the_counter_ \__enumext_level: _tl }
          \tl_put_right:ce { l__enumext_renew_the_count_ \__enumext_level: _tl }
               \exp_not:N \renewcommand { \exp_not:V \l__enumext_ref_the_count_tl } { \exp_not:V \l_
633
        }
634
635
```

Finally the function __enumext_standar_ref: will execute the modification for the reference system in the second argument of the environment definition enumext.

```
636 \cs_new_protected:Nn \__enumext_standar_ref:
637
    {
      \tl_if_empty:cF { l__enumext_renew_the_count_ \__enumext_level: _tl }
638
           \tl_use:c { l__enumext_renew_the_count_ \__enumext_level: _tl }
        }
641
642
```

 $(\textit{End of definition for } \verb|_=enumext_standar_ref:n | and \verb|_=enumext_standar_ref:.)$

13.13.2 Define and set label and ref keys for enumext* and keyans* environments

Here we set the default (*labels*) for enumext* and keyans* environments, along with the default value for

```
labelwidth key and ref key.
 \l__enumext_label_vii_tl
                            643 \cs_set_protected:Npn \__enumext_tmp:nnn #1 #2 #3
\l__enumext_label_viii_tl
                            644
                                  \keys_define:nn { enumext / #1 }
                            645
                                     {
                            646
                                       label .code:n
                                                            \__enumext_label_style:cvn { l__enumext_label_#2_tl }
                                                               { l__enumext_counter_#2_tl } {##1}
                                                            \dim_set_eq:cN { l__enumext_labelwidth_#2_dim }
                                                              \l__enumext_current_widest_dim
                                       label .initial:n = #3,
                                       label .value_required:n = true,
                                       ref
                                             .code:n = \__enumext_starred_ref:n {##1},
                                             .value_required:n = true,
                                       ref
```

}

©2024 by Pablo González L

} 658

657

41/156

```
659 \__enumext_tmp:nnn { enumext* } { vii } { \arabic*.}
660 \__enumext_tmp:nnn { keyans* } { viii } { \Alph*) }
```

(End of definition for label and others.)

__enumext_starred_ref:n
__enumext_starred_ref:

The implementation of __enumext_starred_ref:n is the same as that used for the environment enumext.

```
661 \cs_new_protected:Npn \__enumext_starred_ref:n #1
      \tl_set:Nn \l__enumext_ref_key_arg_tl {#1}
663
      \int_compare:nNnT { \l__enumext_level_h_int } = { 1 }
664
          \tl_if_empty:NTF \l__enumext_ref_key_arg_tl
              \msg_error:nnn { enumext } { key-ref-empty } { enumext* }
              \tl_set_eq:NN \l__enumext_ref_the_count_tl \l__enumext_counter_vii_tl
              \__enumext_regex_counter_style:
              \tl_set_eq:NN \l__enumext_ref_the_count_tl \l__enumext_the_counter_vii_tl
              \tl_put_right:Ne \l__enumext_renew_the_count_vii_tl
                  \exp_not:N \renewcommand { \exp_not:V \l__enumext_ref_the_count_tl } { \exp_not:V
677
678
        }
      \int_compare:nNnT { \l__enumext_keyans_level_h_int } = { 1 }
          \tl_if_empty:NTF \l__enumext_ref_key_arg_tl
            {
              \msg_error:nnn { enumext } { key-ref-empty } { keyans* }
            }
              \tl_set_eq:NN \l__enumext_ref_the_count_tl \l__enumext_counter_viii_tl
              \__enumext_regex_counter_style:
              \tl_set_eq:NN \l__enumext_ref_the_count_tl \l__enumext_the_counter_viii_tl
              \tl_put_right:Ne \l__enumext_renew_the_count_viii_tl
                  \exp_not:N \renewcommand { \exp_not:V \l__enumext_ref_the_count_tl } { \exp_not:`
                }
            }
        }
```

Finally the function __enumext_starred_ref: will execute the modification for the reference system in the second argument of the enumext* and keyans* environment definition.

(End of definition for __enumext_starred_ref:n and __enumext_starred_ref:.)

13.13.3 Define and set label and ref keys for keyans and keyanspic environments

Here we set the default $\langle label \rangle$ for keyans and keyanspic environment, along with the default value for labelwidth and ref key. The keyanspic environment use the same $\langle label \rangle$ as the keyans environment.

```
\__enumext_label_style:cvn { l__enumext_label_v_tl }

{ l__enumext_counter_v_tl } {#1}

\dim_set_eq:cN { l__enumext_labelwidth_v_dim }

\l_enumext_current_widest_dim

\_enumext_label_style:cvn { l__enumext_label_vi_tl }

{ l__enumext_counter_vi_tl } {#1}

\dim_set_eq:cN { l__enumext_labelwidth_v_dim }

\l_enumext_current_widest_dim

},

\label .initial:n = \Alph*),

\label .value_required:n = true,

ref .code:n = \__enumext_keyans_ref:n {#1},

ref .value_required:n = true,

}
```

(End of definition for label and others.)

__enumext_keyans_ref:n
__enumext_keyans_ref:

The implementation of __enumext_keyans_ref:n is the same as that used for the environment enumext.

```
\cs_new_protected:Npn \__enumext_keyans_ref:n #1
732
      \tl_set:Nn \l__enumext_ref_key_arg_tl {#1}
      \tl_if_empty:NTF \l__enumext_ref_key_arg_tl
734
735
        {
           \msg_error:nnn { enumext } { key-ref-empty } { keyans }
736
        }
737
        {
738
           \tl_set_eq:NN \l__enumext_ref_the_count_tl \l__enumext_counter_v_tl
739
           \__enumext_regex_counter_style:
           \tl_set_eq:NN \l__enumext_ref_the_count_tl \l__enumext_the_counter_v_tl
          \tl_put_right:Ne \l__enumext_renew_the_count_v_tl
               \exp_not:N \renewcommand { \exp_not:V \l__enumext_ref_the_count_tl } { \exp_not:V \l_
        }
747
```

Finally the function __enumext_keyans_ref: will execute the modification for the reference system in the second argument of the keyans* environment definition.

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_keyans_ref:n \ and\ \verb|_-enumext_keyans_ref:|)$

13.14 Setting start, start* and widest keys

__enumext_start_from:NNn
__enumext_start_from:ccn
__enumext_start_from:cce

The function $_$ enumext_start_from:NNn used by start and start* keys take three arguments:

```
#1: \l__enumext_label_X_tl
#2: \l__enumext_start_X_int
#3: \langle integer or string \rangle
```

The first argument of this function are the "counter style" set by label key, the second argument is returned by the function, the third argument can be an $\langle integer \rangle$ or $\langle string \rangle$ of the form \Alph , \alph , \alph , \alph , \alph , \alph or \alph . This effectively allows \alph or \alph or \alph to be used.

(End of definition for $\label{lem:nnn} \$) = enumext_start_from:NNn.)

```
\__enumext_widest_from:nNNn
\__enumext_widest_from:nccn
```

widest

The function $_$ _enumext_widest_from:nNNn used by the widest key take four arguments:

‡1: The counter associated with the environment level

#2: \l__enumext_label_X_tl

#3: \l__enumext_labelwidth_X_dim

#4: \langle integer or stringangle

The second and third arguments of this function are the values set by label and labelwidth keys, the four argument can be an $\langle integer \rangle$ or $\langle string \rangle$ of the form \Alph, \alph, \Roman or \roman. The value of the four argument is set temporarily for the identified counter in this point (level), then the value is expanded into a "box" and the "width" of the "box" is returned.

```
769 \cs_new_protected:Npn \__enumext_widest_from:nNNn #1 #2 #3 #4
    {
770
      \__enumext_if_is_int:nTF {#4}
771
          \setcounter{enumX#1} { #4 }
        }
        {
          \regex_match:nVT { \c{Alph} | \c{alph} } {#2}
             { \setcounter{enumX#1} { \int_from_alph:n {#4} } }
          \regex_match:nVT { \c{Roman} | \c{roman} } {#2}
             { \setcounter{enumX#1} { \int_from_roman:n {#4} } }
       \__enumext_label_width_by_box:cv
         { l__enumext_labelwidth_#1_dim } { l__enumext_label_#1_tl }
782
784 \cs_generate_variant:Nn \__enumext_widest_from:nNNn { nccn }
```

($End\ of\ definition\ for\ _enumext_widest_from:nNNn.$)

start Now define and set start*, start and widest keys for enumext, enumext*, keyans and keyans* environments.

```
785 \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
786
      \keys_define:nn { enumext / #1 }
787
788
        {
          start* .code:n
                             = {
                                  \__enumext_start_from:ccn
                                   { l__enumext_label_#2_tl }
                                   { l__enumext_start_#2_int } {##1}
                               },
           start* .value_required:n = true,
           start .code:n
                             = {
                                 \__enumext_start_from:cce
                                   { l__enumext_label_#2_tl }
                                   { l__enumext_start_#2_int } { \int_eval:n {##1} }
                               },
           start .initial:n = 1,
           start
                 .value_required:n = true,
          widest .code:n
                                  \__enumext_widest_from:nccn {#2}
                                   { l__enumext_label_#2_tl }
                                   { l__enumext_labelwidth_#2_dim } {##1}
                               },
          widest .value_required:n = true,
        }
808
810 \clist_map_inline:Nn \c__enumext_all_envs_clist { \__enumext_tmp:nn #1 }
```

(End of definition for start, start*, and widest.)

13.15 Setting keys for vertical spaces

```
Define and set topsep, partopsep, parsep, itemsep, noitemsep and nosep keys for enumext, enumext*,
   topsep
           keyans and keyans* environments.
partopsep
            811 \cs_set_protected:Npn \__enumext_tmp:nnnnnn #1 #2 #3 #4 #5 #6
noitemsep
                {
            812
    nosep
                   \keys_define:nn { enumext / #1 }
            813
                     {
            814
                       topsep
                                  .skip_set:c = { l__enumext_topsep_#2_skip },
           ©2024 by Pablo González L
                                                                                                               44/156
```

```
topsep
                    .initial:n = \{\#3\},
          topsep
                    .value_required:n = true,
817
          partopsep .skip_set:c = { l__enumext_partopsep_#2_skip },
818
          partopsep .initial:n = {#4},
819
          partopsep .value_required:n = true,
          parsep .skip_set:c = { l__enumext_parsep_#2_skip },
821
          parsep .initial:n = {#5},
                   .value_required:n = true,
          parsep
823
          itemsep .skip_set:c = { l__enumext_itemsep_#2_skip },
824
          itemsep .initial:n = {#6},
          itemsep .value_required:n = true,
          noitemsep .meta:n = { itemsep = 0pt, parsep = 0pt },
          noitemsep .value_forbidden:n = true,
828
                   .meta:n = {
          nosep
                                    itemsep = 0pt, parsep= 0pt,
                                    topsep = Opt, partopsep = Opt,
831
832
          nosep
                    .value_forbidden:n = true,
833
834
835
```

Now we set the values based on standard article class in 10pt.

```
836 \__enumext_tmp:nnnnnn { level-1 } { i } { 8.0pt plus 2.0pt minus 4.0pt }
837 { 2.0pt plus 1.0pt minus 1.0pt } { 4.0pt plus 2.0pt minus 1.0pt }
   { 4.0pt plus 2.0pt minus 1.0pt }
_{839} \__enumext_tmp:nnnnnn { level-2 } { ii } { 4.0pt plus 2.0pt minus 1.0pt }
_{840} { 2.0pt plus 1.0pt minus 1.0pt } { 2.0pt plus 1.0pt minus 1.0pt }
   { 2.0pt plus 1.0pt minus 1.0pt }
842 \__enumext_tmp:nnnnnn { level-3 } { iii } { 2.0pt plus 1.0pt minus 1.0pt }
   { 1.0pt minus 1.0pt }{ 0pt }{ 2.0pt plus 1.0pt minus 1.0pt }
844 \__enumext_tmp:nnnnnn { level-4 } { iv } { 2.0pt plus 1.0pt minus 1.0pt }
    { 1.0pt minus 1.0pt }{ 0pt }{ 2.0pt plus 1.0pt minus 1.0pt }
_{8.46} \__enumext_tmp:nnnnnn { keyans } { v }{ 4.0pt plus 2.0pt minus 1.0pt }
   { 2.0pt plus 1.0pt minus 1.0pt }{ 2.0pt plus 1.0pt minus 1.0pt }
   { 2.0pt plus 1.0pt minus 1.0pt }
{ 2.0pt plus 1.0pt minus 1.0pt } { 4.0pt plus 2.0pt minus 1.0pt }
   { 4.0pt plus 2.0pt minus 1.0pt }
852 \__enumext_tmp:nnnnnn { keyans* } { viii } { 4.0pt plus 2.0pt minus 1.0pt }
   { 2.0pt plus 1.0pt minus 1.0pt } { 2.0pt plus 1.0pt minus 1.0pt }
    { 2.0pt plus 1.0pt minus 1.0pt }
```

(End of definition for topsep and others.)

13.16 Setting base-fix key

When nesting starting right after \item (without material between them) there is a problem with the alignment of the *baseline* between the two environments. One way to get around this problem is to place \mode_leave_vertical: apply \vspace{-\baselineskip} and set \topsep=0pt for the "first level" of the nested enumext environment.

base-fix
__enumext_nested_base_line_fix:

We define the key base-fix only for the "first level" of enumext environment.

The function __enumext_nested_base_line_fix: passed to the __enumext_parse_keys:n function in the definition of the enumext environment ($\S13.38$) will be responsible for applying the *baseline correction* and adjusting the $\langle keys \rangle$ for the enumext environment and the \printkeyans with *starred argument* '*' ($\S13.46$)

We will first implement the function code from the user side of the base-fix key, that is, only the user knows when it is necessary to apply it within the document in which case the variable \l_enumext_print_-keyans_star_bool set by the \printkeyans command is false and the variable \l_enumext_base_-line_fix_bool is true.

When we are running the \printkeyans command with the *starred argument* '*' the variable \l_-enumext_print_keyans_star_bool is true and we can run a simplified version of \vspace using \skip_vertical:n.

Finally we set the values of the keys topsep, above and above* for the "first level" of enumext environment equal to opt and set the variable \l_enumext_base_line_fix_bool to false.

```
keys_set:nn { enumext / level-1 }

{
    topsep = Opt, above = Opt, above* = Opt,
    }

bool_set_false:N \l__enumext_base_line_fix_bool
}
```

(End of definition for base-fix and $\ensuremath{\backslash}$ _enumext_nested_base_line_fix:.)

13.17 Setting keys for horizontal spaces

rightmargin listparindent list-offset list-indent

itemindent Define and set itemindent, rightmargin, listparindent, list-offset and list-indent keys for enumext, enumext*, keyans and keyans* environments.

```
887 \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
      \keys_define:nn { enumext / #1 }
        {
          itemindent
                        .dim_set:c = { l__enumext_fake_item_indent_#2_dim },
          itemindent
                        .value required:n = true.
892
          rightmargin .dim_set:c = { l__enumext_rightmargin_#2_dim },
893
          rightmargin
                        .value_required:n = true,
894
          listparindent .dim_set:c = { l__enumext_listparindent_#2_dim },
895
          listparindent .value_required:n = true,
          list-offset
                         .dim_set:c = { l__enumext_listoffset_#2_dim },
          list-offset
                         .value_required:n = true,
          list-indent
                         .code:n
                           \bool_set_true:c { l__enumext_leftmargin_tmp_#2_bool }
                           \dim_set:cn { l__enumext_leftmargin_tmp_#2_dim } {##1},
          list-indent
                        .value_required:n = true,
        }
903
    }
904
905 \clist_map_inline:nn
    {
      {level-1}{i}, {level-2}{ii}, {level-3}{iii}, {level-4}{iv}, {keyans}{v}
907
    }
    { \__enumext_tmp:nn #1 }
```

 $(End\ of\ definition\ for\ {\tt itemindent}\ \ and\ others.)$

For enumext* and keyans* environments the situation is a bit different, the list-indent key behaves like the list-offset key.

```
gno \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
gno \cs_set_protected:Npn \__enumext_fake_item_indent gno \cs_set_protected
gno \cs_set_protected:Npn \__enumext_fake_item_indent_#2_dim },
gno \cs_set_protected:Npn \_enumext_fake_item_indent_#2_dim },
gno \cs_set_protected:Npn \cs_set_protected:
```

```
list-offset
                        .dim_set:c = { l__enumext_listoffset_#2_dim },
                        .value_required:n = true,
          list-offset
                                 = { list-offset = ##1 },
          list-indent
                        .meta:n
          list-indent
                        .value_required:n = true,
    }
925
  \clist_map_inline:nn
    {
      {enumext*}{vii}, {keyans*}{viii}
    }
      \__enumext_tmp:nn #1 }
```

13.17.1 Functions for setting the fake itemindent

__enumext_fake_item_indent:
 __enumext_keyans_fake_item_indent:
 __enumext_fake_item_indent_vii:
 __enumext_fake_item_indent_viii:

The itemindent key does not set the value of \itemindent, it only sets the value of the *horizontal space* applied using \skip_horizontal:N. We will store this value in the variable and only apply it when it is greater than opt. Here I will need to place \mode_leave_vertical: and the plain TeX macro \ignorespaces to avoid unwanted extra space when using the itemindent key.

```
931 \cs_set_protected:Nn \__enumext_fake_item_indent:
932
      \dim_compare:nNnT
933
        { \dim_use:c { l__enumext_fake_item_indent_ \__enumext_level: _dim } }
        { \c_zero_dim }
        {
          \tl_set:ce { l__enumext_fake_item_indent_ \__enumext_level: _tl }
               \exp_not:N \mode_leave_vertical:
               \exp_not:n { \skip_horizontal:n }
                 { \dim_use:c { l__enumext_fake_item_indent_ \__enumext_level: _dim } }
               \exp_not:N \ignorespaces
        }
     }
  \cs_set_protected:Nn \__enumext_keyans_fake_item_indent:
948
      \dim_compare:nNnT
        { \l__enumext_fake_item_indent_v_dim } > { \c_zero_dim }
         {
951
           \tl_set:Ne \l__enumext_fake_item_indent_v_tl
             {
               \exp_not:N \mode_leave_vertical:
               \exp_not:N \skip_horizontal:N \l__enumext_fake_item_indent_v_dim
               \exp_not:N \ignorespaces
        }
058
959
  \cs_set_protected:Nn \__enumext_fake_item_indent_vii:
960
    {
961
      \dim compare:nNnT
962
         { \l__enumext_fake_item_indent_vii_dim } > { \c_zero_dim }
963
           \tl_set:Ne \l__enumext_fake_item_indent_vii_tl
               \exp_not:N \skip_horizontal:N \l__enumext_fake_item_indent_vii_dim
               \exp_not:N \ignorespaces
             }
        }
970
     }
971
972 \cs_set_protected:Nn \__enumext_fake_item_indent_viii:
973
      \dim_compare:nNnT
974
        { \l__enumext_fake_item_indent_viii_dim } > { \c_zero_dim }
          \tl_set:Ne \l__enumext_fake_item_indent_viii_tl
078
               \exp_not:N \skip_horizontal:N \l__enumext_fake_item_indent_viii_dim
               \exp_not:N \ignorespaces
981
        }
     }
983
```

(End of definition for __enumext_fake_item_indent: and others.)

13.18 Setting show-length key

show-length

after

first

Define and set show-length key for enumext, enumext*, keyans and keyans* environments. The function sets the boolean variable \l__enumext_show_length_X_bool used in the definition of all environments to "true" and calls the function __enumext_show_length:nnn which prints all the values of the "vertical" and "horizontal" parameters calculated and used.

(End of definition for show-length.)

13.19 Setting before, after and first keys

before Define and set before, before*, after and first keys for enumext, enumext*, keyans and keyans* before* environments.

```
993 \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
994
    {
       \keys_define:nn { enumext / #1 }
995
996
           before .tl_set:c = { l__enumext_before_no_starred_key_#2_tl },
997
           before .value_required:n = true,
998
           before* .tl_set:c = { l__enumext_before_starred_key_#2_tl },
          before* .value_required:n = true,
                  .tl_set:c = { l__enumext_after_stop_list_#2_tl },
          after
                  .value_required:n = true,
          after
          first
                  .tl_set:c = { l__enumext_after_list_args_#2_tl },
           first
                 .value_required:n = true,
1004
1005
    }
1007 \clist_map_inline:Nn \c__enumext_all_envs_clist { \__enumext_tmp:nn #1 }
```

(End of definition for before and others.)

13.19.1 Functions for before, after and first keys in enumext

__enumext_before_args_exec:
__enumext_before_keys_exec:
__enumext_after_stop_list:
__enumext_after_args_exec:

The function __enumext_before_args_exec: executes the $\{\langle code \rangle\}$ set by the before* key "before" the enumext environment is started. The $\{\langle code \rangle\}$ is executed "without" knowing any definition of the $\{\langle arg\ two \rangle\}$ of the list: $\{\langle code \rangle\}$ \list $\{\langle arg\ one \rangle\}$ $\{\langle arg\ two \rangle\}$.

```
1008 \cs_new_protected:Nn \__enumext_before_args_exec:
1009 {
1010 \tl_use:c { l__enumext_before_starred_key_ \__enumext_level: _tl }
1011 }
```

The function __enumext_before_keys_exec: executes the $\{\langle code \rangle\}$ set by the before key "before" the enumext environment is started in second argument of the list. The $\{\langle code \rangle\}$ is executed "knowing" all definition and values provides by $\langle keys \rangle$: \list $\{\langle arg\ one \rangle\}\{\langle arg\ two \rangle\}\{\langle code \rangle\}\}$

```
1012 \cs_new_protected:Nn \__enumext_before_keys_exec:
1013 {
1014 \tl_use:c { l__enumext_before_no_starred_key_ \__enumext_level: _tl }
```

The function __enumext_after_stop_list: executes the $\{\langle code \rangle\}$ set by the after key "after" the enumext environment has finished: $\end{list}\{\langle code \rangle\}$.

```
1016 \cs_new_protected:Nn \__enumext_after_stop_list:
1017 {
1018 \tl_use:c { l__enumext_after_stop_list_ \__enumext_level: _tl }
1019 }
```

The function __enumext_after_args_exec: executes the $\{\langle code \rangle\}$ set by the first key after the end of the second argument of the list defining the enumext environment, just before the first occurrence of \item: \list{\lang one}\}{\langle code}\}\item.

48 / 156

 $(\textit{End of definition for } \verb|_enumext_before_args_exec: and others.)$

13.19.2 Functions for before, after and first keys in keyans

```
Same implementation as the one used in the enumext environment.
\__enumext_before_args_exec_v:
\ enumext before kevs exec v:
                         \cs_new_protected:Nn \__enumext_before_args_exec_v:
 \__enumext_after_stop_list_v:
\__enumext_after_args_exec_v:
                                 \tl_use:N \l__enumext_before_starred_key_v_tl
                         \cs_new_protected:Nn \__enumext_before_keys_exec_v:
                                 \tl_use:N \l__enumext_before_no_starred_key_v_tl
                         1031
                             \cs_new_protected:Nn \__enumext_after_stop_list_v:
                         1032
                                 \tl_use:N \l__enumext_after_stop_list_v_tl
                         1034
                             \cs_new_protected:Nn \__enumext_after_args_exec_v:
                                 \tl_use:N \l__enumext_after_list_args_v_tl
```

(End of definition for __enumext_before_args_exec_v: and others.)

13.19.3 Functions for before, after and first keys in enumext* and keyans*

__enumext_before_args_exec_vii:
__enumext_before_keys_exec_vii
__enumext_after_stop_list_vii:
__enumext_after_args_exec_vii:

Same implementation as the one used in the enumext environment.

```
\cs_new_protected:Nn \__enumext_before_args_exec_vii:
       \tl_use:N \l__enumext_before_starred_key_vii_tl
1043
   \cs_new_protected:Nn \__enumext_before_args_exec_viii:
1045
       \tl_use:N \l__enumext_before_starred_key_viii_tl
   \cs_new_protected:Nn \__enumext_before_keys_exec_vii:
       \tl_use:N \l__enumext_before_no_starred_key_vii_tl
1051
   \cs_new_protected:Nn \__enumext_before_keys_exec_viii:
1053
       \tl_use:N \l__enumext_before_no_starred_key_viii_tl
1054
1055
   \cs_new_protected:Nn \__enumext_after_stop_list_vii:
1057
       \tl_use:N \l__enumext_after_stop_list_vii_tl
1058
\cs_new_protected:Nn \__enumext_after_stop_list_viii:
       \tl_use:N \l__enumext_after_stop_list_viii_tl
1063
\cs_new_protected:Nn \__enumext_after_args_exec_vii:
1065
       \tl_use:N \l__enumext_after_list_args_vii_tl
1066
   \cs_new_protected:Nn \__enumext_after_args_exec_viii:
       \tl_use:N \l__enumext_after_list_args_viii_tl
```

(End of definition for $\ensuremath{\backslash}$ enumext_before_args_exec_vii: and others.)

1076

mini-env

13.20 Setting keys for multicols and minipage

mini-env mini-sep The default value of the columns-sep key is handled by the state of the boolean variable \l__enumext_- columns_sep_X_bool which is handled in the internal definition of the enumext and keyans environments.

Define and set mini-env, mini-sep, columns-sep and columns keys for enumext, enumext*, keyans and keyans* environments.

1072 \cs_set_protected:Npn __enumext_tmp:nn #1 #2
1073 {
1074 \keys_define:nn { enumext / #1 }

©2024 by Pablo González L 49/156

.dim_set:c = { l__enumext_minipage_right_#2_dim },

```
mini-env
                        .value_required:n = true,
                       .dim_set:c = { l__enumext_minipage_hsep_#2_dim },
           mini-sen
                       .initial:n = 0.3333em,
           mini-sep
                       .value_required:n = true,
           mini-sep
           columns-sep .dim_set:c = { l__enumext_columns_sep_#2_dim },
1081
           columns-sep .value_required:n = true,
1082
           columns
                       .int_set:c = { l__enumext_columns_#2_int },
1083
           columns
                       .initial:n = 1,
           columns
                       .value_required:n = true,
1088 \clist_map_inline:Nn \c__enumext_all_envs_clist { \__enumext_tmp:nn #1 }
```

For enumext* and keyans* environments the situation is a bit different, the command \miniright is not available, so we will add the keys mini-right and mini-right* to implement support for minipage environment.

```
1089 \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
     {
1090
       \keys_define:nn { enumext / #1 }
1091
1092
           mini-right .tl_gset:c = { g__enumext_miniright_code_#2_tl },
1093
           mini-right .value_required:n = true,
           mini-right* .code:n
                                       \bool_gset_true:c { g__enumext_minipage_center_#2_bool }
                                      \keys_set:nn { enumext / #1 } { mini-right = {##1} }
                                    Դ.
           mini-right* .value_required:n = true,
         }
1100
1101
1102 \clist_map_inline:nn { {enumext*}{vii}, {keyans*}{viii} } { \__enumext_tmp:nn #1 }
```

(End of definition for mini-env and others.)

13.21 Adjustment of vertical spaces for multicols

When nesting a "list environment" inside the multicols environment, the values of the "vertical spaces" are lost, basically the multicols environment takes control over them. Graphically it can be seen like in the figure 7.



Figure 7: Representation of the vertical space in multicols for a nested level.

To keep the desired spaces *above* and *below* in the "list environment" (\topsep + [\partopsep]) it is necessary to "adjust" the spaces added by the multicols environment. The most appropriate option in this case is to use a "context sensitive" vertical space with \addvspace.

I should make it clear that the implementation here is a "bit questionable". At first glance doing \multicolsep=\topsep seemed right, but the results were not always as expected. An almost imperceptible detail is that in some cases the \itemsep values of are "stretched", possibly due to the use of \raggedcolumns and this affects the lower space when closing the environment, which is "smaller" than expected. My attempts to find the correct values using \showoutput and \showboxdepth absolutely failed.

13.21.1 Adjustment of vertical spaces for multicols in enumext

__enumext_multi_set_vskip:

The function __enumext_multi_set_vskip: will take care of determining the "adjusted spaces" that we will apply "above" and "below" the multicols environment in enumext.

We will set the default values taking into account that $T_{\overline{E}}X$ is in $\langle horizontal\ mode \rangle$, then we will make the settings for the $\langle vertical\ mode \rangle$ in which $\langle partopsep \rangle$ comes into play.

Set the values of \l__enumext_multicols_above_X_skip and \l__enumext_multicols_below_X_skip equal to the value of \topsep in the *current level*.

```
1103 \cs_new_protected:Nn \__enumext_multi_set_vskip:
1104 {
1105 \skip_set:cn { l__enumext_multicols_above_ \__enumext_level: _skip }
1106 {
1107 \skip_use:c { l__enumext_topsep_ \__enumext_level: _skip }
1108 }
1109 \skip_set:cn { l__enumext_multicols_below_ \__enumext_level: _skip }
1100 {
```

__enumext_add_pre_parsep:

The function $_$ _enumext_add_pre_parsep: "adjusted" the value of $_$ _enumext_multicols_above_-X_skip detecting the value of $_$ parsep from the previous level. This is necessary since $_$ parsep from the previous level affects the vertical spaces.

```
\cs_new_protected:Nn \__enumext_add_pre_parsep:
1116
       \int_case:nn { \l__enumext_level_int }
           { 2 }{
                   \skip_if_eq:nnF { \l__enumext_parsep_i_skip } { \c_zero_skip }
                       \skip_add:Nn \l__enumext_multicols_above_ii_skip
                            \l__enumext_parsep_i_skip
                     }
                }
           { 3 }{
                   \skip_if_eq:nnF { \l__enumext_parsep_ii_skip } { \c_zero_skip }
                     {
                       \skip_add:Nn \l__enumext_multicols_above_iii_skip
                         {
                           \l__enumext_parsep_ii_skip
1134
1136
           { 4 }{
                   \skip_if_eq:nnF { \l__enumext_parsep_iii_skip } { \c_zero_skip }
1138
                       \skip_add:Nn \l__enumext_multicols_above_iv_skip
                           \l__enumext_parsep_iii_skip
                     }
1144
                }
1145
         }
1146
1147
```

 $(End\ of\ definition\ for\ \verb|_enumext_add_pre_parsep:.)$

__enumext_multi_addvspace:

The function __enumext_multi_addvspace: will apply the spaces set using \addvspace "above" the multicols environment in enumext, taking into account whether TEX is in $\langle horizontal\ mode \rangle$ or $\langle vertical\ mode \rangle$.

```
\cs_new_protected:Nn \__enumext_multi_addvspace:
1149
       \__enumext_multi_set_vskip:
1150
       \mode_if_vertical:T
         {
           \skip_add:cn { l__enumext_multicols_above_ \__enumext_level: _skip }
               \skip_use:c { l__enumext_partopsep_ \__enumext_level: _skip }
           \skip_add:cn { l__enumext_multicols_below_ \__enumext_level: _skip }
               \skip_use:c { l__enumext_partopsep_ \__enumext_level: _skip }
1159
             3
1160
1161
       \par\nopagebreak
1162
       \addvspace{ \skip_use:c { l__enumext_multicols_above_ \__enumext_level: _skip } }
1163
1164
```

(End of definition for __enumext_multi_addvspace:.)

13.21.2 Adjustment of vertical spaces for multicols in keyans

__enumext_keyans_multi_set_vskip:
__enumext_keyans_multi_addvspace:

The function __enumext_keyans_multi_set_vskip: will take care of determining the "adjusted spaces" that we will apply "above" and "below" the multicols environment in keyans. The implementation of this function is the same as the one used in enumext.

```
\cs_new_protected:Nn \__enumext_keyans_multi_set_vskip:
       \skip_set:Nn \l__enumext_multicols_above_v_skip
1168
               _enumext_topsep_v_skip
1160
         }
       \skip set:Nn \l enumext multicols below v skip
         {
            \l__enumext_topsep_v_skip
   \cs_new_protected:Nn \__enumext_keyans_multi_addvspace:
         _enumext_keyans_multi_set_vskip:
1178
       \mode_if_vertical:T
         {
1180
           \skip_add:Nn \l__enumext_multicols_above_v_skip
1181
             {
1182
               \skip_use:N \l__enumext_partopsep_v_skip
1182
1184
           \skip_add:Nn \l__enumext_multicols_below_v_skip
               \skip_use:N \l__enumext_partopsep_v_skip
       \par\nopagebreak
       \addvspace{ \l__enumext_multicols_above_v_skip }
1191
```

 $(End\ of\ definition\ for\ \verb|_enumext_keyans_multi_set_vskip:\ and\ \verb|_enumext_keyans_multi_addvspace:|)$

13.22 Adjustment of vertical spaces for minipage

When nesting a "*list environment*" within the minipage environment, the values of the "*vertical spaces*" are lost. Graphically it can be seen like in the figure 8.



Figure 8: Representation of the minipage spacing adjustment for a nested level.

Since we want to keep the "left" and "right" environments "aligned on top", preserving the \baselineskip and keep the desired "spaces" (\topsep + [\partopsep]) it is necessary to "adjust" the "vertical spaces" for minipage environments.

Here there are several complications that we must circumvent, the minipage environment eliminates the "top" spaces, the multicols environment can be nested in the minipage environment, the "top" and "bottom" spaces are affected when topsep=%pt and to this is added the \partopsep parameter that comes into action according to whether TEX is in \(\lambda \text{horizontal mode} \rangle \text{ vertical mode} \rangle. \) Depending on these cases, small adjustments must be made using \vspace and \addvspace to obtain the "desired vertical spacing".

Again I must make clear that the implementation here is a "bit questionable", but hunting the spaces (glue) produced by the minipage environment is quite complicated, even more if multicols it is nested. The setting of the values was more "trial and error" (aprox to \strutbox), using the help of the lua-visual-debug[14] package, again my attempts to find the correct values using \showoutput and \showboxdepth absolutely failed.

13.22.1 Adjustment of vertical spaces for minipage in enumext

__enumext_minipage_set_skip:
\ enumext minipage add space:

The function __enumext_minipage_set_skip: will take care of determining the "adjust" spaces that we will apply "above" and "below" the __enumext_mini_page environment in enumext.

First we will set the value of $\l_enumext_minipage_right_skip$ equal to $\t_ext_minipage_right_skip$ equal to $\t_ext_minipage_minipage_minipage_after_skip$.

```
1193 \cs_new_protected:Nn \__enumext_minipage_set_skip:
1194 {
1195 \skip_set:Nn \l_enumext_minipage_right_skip
©2024 by Pablo González L
```

We will adjust the values \l__enumext_multicols_above_X_skip and \l__enumext_multicols_below_X_skip and call the function __enumext_pre_itemsep_skip:.

```
\skip_set_eq:cN
\[ \langle \la
```

If the environment multicols is active, we set \topskip=0pt and then we make \multicolsep have the same value as \l_enumext_multicols_above_X_skip.

The function __enumext_minipage_add_space: will apply the spaces on the "left side" using \addvspace "above" the __enumext_mini_page environment, taking into account whether TeX is in \(\lambda \) horizontal mode \(\rangle \) or \(\sqrt{ertical mode} \). Here we use the plain TeX macro \(\cdot{nointerlineskip} \) to prevent baseline "glue" being added between the next pair of boxes in a vertical list. For the latter we will make some adjustments since the \(\rangle \) partopsep parameter comes into play and this affects the vertical spacing.

```
\cs_new_protected:Nn \__enumext_minipage_add_space:
    {
1220
      \__enumext_minipage_set_skip:
      \__enumext_unskip_unkern:
      \mode_if_vertical:TF
        {
         \nopagebreak\nointerlineskip
       }
1226
        {
         \par\nopagebreak\nointerlineskip
1228
         \skip_zero:c { l__enumext_partopsep_ \__enumext_level: _skip }
      \int_compare:nNnTF
       {
         \addvspace{ 0.445\box_ht:N \strutbox }
       }
         \addvspace{ 0.250\box_ht:N \strutbox }
       }
1238
1239
```

 $(\textit{End of definition for } \verb|_=enumext_minipage_set_skip: and \verb|_=enumext_minipage_add_space:|)$

__enumext_pre_itemsep_skip:

The function __enumext_pre_itemsep_skip: will adjust the spaces below the environment minipage and the environment multicols if it is nested in it, taking into account the value of \itemsep from the previous level.

```
}
                    {
                       \dim compare:nNnT
                         { \l__enumext_itemsep_i_skip } < { \l__enumext_minipage_after_skip }
1253
                           \skip_sub:Nn
                             \l__enumext_minipage_after_skip { \l__enumext_itemsep_i_skip }
                             \l__enumext_multicols_below_ii_skip { \l__enumext_itemsep_i_skip }
                           \skip_add:Nn
                             \l__enumext_minipage_after_skip { 0.150\box_ht:N \strutbox }
                           \skip_add:Nn
                             \l__enumext_multicols_below_ii_skip { 0.350\box_ht:N \strutbox }
                       \dim_compare:nNnT
1264
                         { \l__enumext_itemsep_i_skip } > { \l__enumext_minipage_after_skip }
1265
                         {
1266
                           \skip_set:Nn \l__enumext_minipage_temp_skip
1267
                               \l__enumext_itemsep_i_skip - \l__enumext_minipage_after_skip
                           \skip_sub:Nn
                             \l__enumext_minipage_after_skip { \l__enumext_itemsep_i_skip }
                           \skip sub:Nn
                             \l__enumext_multicols_below_ii_skip { \l__enumext_itemsep_i_skip }
1274
                           \skip add:Nn
                             \l__enumext_minipage_after_skip
                             { 0.150\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
1278
                             \l__enumext_multicols_below_ii_skip
                             { 0.350\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
                         }
                    }
                }
1282
           { 3 }{
1284
                  \skip_if_eq:nnTF
1285
                    { \l__enumext_itemsep_ii_skip } { \c_zero_skip }
1286
1287
                       \skip_set:Nn \l__enumext_minipage_after_skip { 0.150\box_ht:N \strutbox }
1288
                       \skip_set:Nn \l__enumext_multicols_below_iii_skip { 0.350\box_ht:N \strutbox }
1289
                    {
                       \dim_compare:nNnT
                         { \l__enumext_itemsep_ii_skip } < { \l__enumext_minipage_after_skip }
                           \skip sub:Nn
                             \l__enumext_minipage_after_skip { \l__enumext_itemsep_ii_skip }
1296
                           \skip_sub:Nn
1297
                             \l__enumext_multicols_below_iii_skip { \l__enumext_itemsep_ii_skip }
1298
                           \skip add:Nn
                             \l__enumext_minipage_after_skip { 0.150\box_ht:N \strutbox }
                           \skip_add:Nn
                             \l__enumext_multicols_below_iii_skip { 0.350\box_ht:N \strutbox }
                       \dim_compare:nNnT
                         { \l__enumext_itemsep_ii_skip } > { \l__enumext_minipage_after_skip }
                         {
1306
                           \skip_set:Nn \l__enumext_minipage_temp_skip
1307
1308
                               \l__enumext_itemsep_ii_skip - \l__enumext_minipage_after_skip
1309
                           \skip_sub:Nn
                             \l__enumext_minipage_after_skip { \l__enumext_itemsep_ii_skip }
                           \skip_sub:Nn
                             \l__enumext_multicols_below_iii_skip { \l__enumext_itemsep_ii_skip }
                           \skip add:Nn
                             \l enumext minipage after skip
1316
                             { 0.150\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
                           \skip_add:Nn
                             \l__enumext_multicols_below_iii_skip
                             { 0.350\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
1320
```

```
}
                }
           { 4 }{
                  \skip_if_eq:nnTF { \l__enumext_itemsep_iii_skip } { \c_zero_skip }
1326
                       \skip_set:Nn \l__enumext_minipage_after_skip { 0.150\box_ht:N \strutbox }
1327
                       \skip_set:Nn \l__enumext_multicols_below_iv_skip { 0.350\box_ht:N \strutbox }
                       \dim compare:nNnT
                         { \l__enumext_itemsep_iii_skip } < { \l__enumext_minipage_after_skip }
                           \skip_sub:Nn
                             \l__enumext_minipage_after_skip { \l__enumext_itemsep_iii_skip }
                           \skip_sub:Nn
                             \l__enumext_multicols_below_iv_skip { \l__enumext_itemsep_iii_skip }
                           \skip_add:Nn
1338
                             \l__enumext_minipage_after_skip { 0.150\box_ht:N \strutbox }
                           \skip_add:Nn
1340
                             \l__enumext_multicols_below_iv_skip { 0.350\box_ht:N \strutbox }
                      \dim_compare:nNnT
                        { \l_enumext_itemsep_iii_skip } > { \l_enumext_minipage_after_skip }
                         {
                           \skip_set:Nn \l__enumext_minipage_temp_skip
1347
                               \l__enumext_itemsep_iii_skip - \l__enumext_minipage_after_skip
1348
1349
                          \skip_sub:Nn
                             \l__enumext_minipage_after_skip { \l__enumext_itemsep_iii_skip }
1351
                           \skip_sub:Nn
                             \l__enumext_multicols_below_iv_skip { \l__enumext_itemsep_iii_skip }
                           \skip_add:Nn
                             \l__enumext_minipage_after_skip
                             { 0.150\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
                           \skip_add:Nn
                             \l__enumext_multicols_below_iv_skip
1358
                             { 0.350\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
                         }
1360
                    }
                }
         }
1363
```

(End of definition for __enumext_pre_itemsep_skip:.)

13.22.2 Adjustment of vertical spaces for minipage in keyans

__enumext_keyans_minipage_set_skip:
__enumext_keyans_minipage_add_space:
__enumext_keyans_pre_itemsep_skip:

The function __enumext_keyans_mini_set_vskip: will take care of determining the "adjusted" spaces that we will apply "above" and "below" the __enumext_mini_page environment in keyans. The implementation of this function is the same as the one used in enumext.

```
1365 \cs_new_protected:Nn \__enumext_keyans_minipage_set_skip:
1366
      \skip_zero:N \l__enumext_minipage_after_skip
1367
      \skip_zero:N \l__enumext_minipage_left_skip
      \skip_zero:N \l__enumext_minipage_right_skip
      \skip_set:Nn \l__enumext_minipage_right_skip
            _enumext_topsep_v_skip
        }
      \mode_if_vertical:T
          \skip_add:Nn \l__enumext_minipage_right_skip
             \l__enumext_partopsep_v_skip
1378
      1381
      \skip_set_eq:NN \l__enumext_multicols_above_v_skip \l__enumext_minipage_right_skip
1382
      \skip_set_eq:NN \l__enumext_multicols_below_v_skip \l__enumext_minipage_right_skip
1383
      \__enumext_keyans_pre_itemsep_skip:
1384
```

```
\int_compare:nNnT { \l__enumext_columns_v_int } > { 1 }
            \skip zero:N \topskip
1387
           \skip_set_eq:NN \multicolsep \l__enumext_minipage_right_skip
1388
1389
     }
1390
   \cs_new_protected:Nn \__enumext_keyans_minipage_add_space:
1391
1392
       \__enumext_keyans_minipage_set_skip:
1393
       \__enumext_unskip_unkern:
1394
       \mode_if_vertical:TF
         {
           \nopagebreak\nointerlineskip
1397
         }
1398
         {
1399
            \par\nopagebreak\nointerlineskip
1400
            \skip_zero:N \l__enumext_partopsep_v_skip
1401
1402
       \int_compare:nNnTF { \l__enumext_columns_v_int } > { 1 }
1403
            \addvspace{ 0.445\box_ht:N \strutbox }
         }
         {
            \addvspace{ 0.250\box_ht:N \strutbox }
1408
1409
1410
   \cs_new_protected:Nn \__enumext_keyans_pre_itemsep_skip:
1411
1412
       \skip_if_eq:nnTF
1413
         { \l__enumext_itemsep_i_skip } { \l__enumext_minipage_after_skip }
1414
1415
           \skip_set:Nn \l__enumext_minipage_after_skip { 0.150\box_ht:N \strutbox }
           \skip_set:Nn \l__enumext_multicols_below_v_skip { 0.350\box_ht:N \strutbox }
         }
1418
         {
           \dim compare:nNnT
1420
             { \l__enumext_itemsep_i_skip } < { \l__enumext_minipage_after_skip }
1421
1422
                \skip_sub:Nn \l__enumext_minipage_after_skip { \l__enumext_itemsep_i_skip }
1423
                \skip_sub:Nn \l__enumext_multicols_below_v_skip { \l__enumext_itemsep_i_skip }
                \skip_add:Nn \l__enumext_minipage_after_skip { 0.150\box_ht:N \strutbox }
                \skip_add:Nn \l__enumext_multicols_below_v_skip { 0.350\box_ht:N \strutbox }
           \dim_compare:nNnT
             { \l__enumext_itemsep_i_skip } > { \l__enumext_minipage_after_skip }
             {
                \skip_set:Nn \l__enumext_minipage_temp_skip
1431
                  {
                    \l__enumext_itemsep_i_skip - \l__enumext_minipage_after_skip
1433
                  }
                \skip_sub:Nn \l__enumext_minipage_after_skip { \l__enumext_itemsep_i_skip }
                \skip_sub:Nn \l__enumext_multicols_below_v_skip { \l__enumext_itemsep_i_skip }
                \skip_add:Nn \l__enumext_minipage_after_skip
                  { 0.150\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
                \skip_add:Nn \l__enumext_multicols_below_v_skip
                  { 0.350\box_ht:N \strutbox + \l__enumext_minipage_temp_skip }
1441
        }
1442
1443
```

(End of definition for __enumext_keyans_minipage_set_skip:, __enumext_keyans_minipage_add_space:, and __enumext keyans pre itemsep skip:)

13.22.3 Adjustment of vertical spaces for minipage in enumext* and keyans*

__enumext_mini_set_vskip_vii:
__enumext_mini_set_vskip_viii:

The functions __enumext_mini_set_vskip_vii: and __enumext_mini_set_vskip_viii: will take care of determining the "adjusted" spaces that we will apply "above" and "below" the __enumext_mini_page environment in enumext* and keyans*.

```
1444 \cs_new_protected:Nn \__enumext_mini_set_vskip_vii:
1445 {
1446 \skip_zero_new:N \l__enumext_minipage_left_skip
1447 \skip_gzero_new:N \g__enumext_minipage_right_skip
```

©2024 by Pablo González L

56 / 156

```
\skip_gzero_new:N \g__enumext_minipage_after_skip
        \skip_if_eq:nnTF { \l__enumext_topsep_vii_skip } { \c_zero_skip }
            \skip_set:Nn \l__enumext_minipage_left_skip { 0.5\box_dp:N \strutbox }
1451
            \skip_gset:Nn \g__enumext_minipage_right_skip { 0.325\box_dp:N \strutbox }
          }
1453
          {
1454
            \skip_set:Nn \l__enumext_minipage_left_skip { 0.5875\box_dp:N \strutbox }
1455
            \skip_gset:Nn \g__enumext_minipage_right_skip
                \l__enumext_topsep_vii_skip
              3
            \skip_gset:Nn \g__enumext_minipage_after_skip
              {
                0.325\box_dp:N \strutbox + \l__enumext_topsep_vii_skip
1462
1463
          }
1464
1465
   \cs_new_protected:Nn \__enumext_mini_set_vskip_viii:
1466
1467
        \skip_zero_new:N \l__enumext_minipage_after_skip
        \skip_zero_new:N \l__enumext_minipage_left_skip
        \skip_zero_new:N \l__enumext_minipage_right_skip
        \skip_if_eq:nnTF { \l__enumext_topsep_viii_skip } { \c_zero_skip }
1471
            \skip_set:Nn \l__enumext_minipage_left_skip
1473
              {
1474
                0.5\box_dp:N \strutbox
1475
              }
            \skip_set:Nn \l__enumext_minipage_right_skip
                \l__enumext_partopsep_viii_skip
              3
            \skip_set:Nn \l__enumext_minipage_after_skip
1481
              {
1482
                1.6\box_dp:N \strutbox
1483
1484
          }
1485
          {
1486
            \skip_set:Nn \l__enumext_minipage_left_skip
1487
                0.5875\box_dp:N \strutbox
            \skip_set:Nn \l__enumext_minipage_right_skip
                \l__enumext_topsep_viii_skip
1494
            \skip_set:Nn \l__enumext_minipage_after_skip
1495
1496
                0.325\box_dp:N \strutbox + \l__enumext_topsep_viii_skip
1497
              }
           }
     }
(\mathit{End}\ of\ definition\ for\ \verb|\_enumext_mini_set_vskip_vii:\ and\ \verb|\_enumext_mini_set_vskip_viii:)
```

__enumext_mini_addvspace_vii: __enumext_mini_addvspace_viii:

The functions __enumext_mini_addvspace_vii: and __enumext_mini_addvspace_viii: will apply the vertical space "only above" the __enumext_mini_page environment on the left side when the mini-right key is active in the enumext* and keyans* environments.

Here we will NOT take into account whether TeX is in \(\lambda \text{orizontal mode} \) or \(\text{vertical mode} \rangle \), since \(\text{partopsep} \) is equal to opt in both environments.

```
\cs_new_protected:Nn \__enumext_mini_addvspace_vii:
        __enumext_mini_set_vskip_vii:
       \par\nopagebreak
       \addvspace { \l__enumext_minipage_left_skip }
     }
1506
   \cs_new_protected:Nn \__enumext_mini_addvspace_viii:
1507
1508
       \__enumext_mini_set_vskip_viii:
1509
       \par\nopagebreak
©2024 by Pablo González L
```

```
\land \addvspace { \l__enumext_minipage_left_skip }
\[ \land of definition for \__enumext_mini_addvspace_vii: and \__enumext_mini_addvspace_viii:.)
```

13.22.4 The command \miniright

The command \miniright will close the __enumext_mini_page environment on the "left side", open the __enumext_mini_page environment on the "right side" adding the adjusted vertical space. By default we will add \centering when starting the "right side" environment. The starred argument '*' inhibits the use of \centering command i.e. the usual ETEX justification is maintained in the __enumext_mini_page on the "right side".

\miniright First we will perform some checks to prevent the command from being executed outside the enumext environment or somewhere inappropriate then we will call the internal functions to execute it in the enumext and keyans environments.

```
NewDocumentCommand \miniright { s }
       \int_compare:nNnT { \l__enumext_keyans_pic_level_int } = { 1 }
         {
           \msg_error:nnn { enumext } { wrong-miniright-place }
         }
1518
       % outside
1519
       \bool_lazy_and:nnT
         { \int_compare_p:nNn { \l__enumext_level_int } = { 0 } }
         { \int_compare_p:nNn { \l__enumext_level_h_int } = { 0 } }
           \msg_error:nnn { enumext } { wrong-miniright-place }
         }
       % starred env
       \bool_if:NT \l__enumext_starred_bool
         {
1528
           \msg_error:nnn { enumext } { wrong-miniright-starred }
       \int_compare:nNnTF { \l__enumext_keyans_level_int } = { 1 }
             _enumext_keyans_mini_right_cmd:n {#1}
             _enumext_mini_right_cmd:n {#1} }
1536
```

(End of definition for \miniright. This function is documented on page 11.)

©2024 by Pablo González L

__enumext_mini_right_cmd:n

The function __enumext_mini_right_cmd:n takes as argument the *starred* '*' of the \miniright command in the enumext environment. We check if the mini-env key is active via the variable \l__enumext_-minipage_right_X_dim, if so we close the multicols environment with the __enumext_mini_page environment on the "left side", then we open the __enumext_mini_page environment on the "right side", apply our adjusted "vertical spaces", followed by adding the \centering command when the starred argument '*' is not present and set zero \g__enumext_minipage_stat_int, otherwise we return an error.

```
\cs_new_protected:Npn \__enumext_mini_right_cmd:n #1
1537
       \dim compare:nNnTF
         { \dim_use:c { l__enumext_minipage_right_ \__enumext_level: _dim } } > { \c_zero_dim }
         {
           \__enumext_multicols_stop:
           \int_compare:nNnT
1543
             { \int_use:c { l__enumext_columns_ \__enumext_level: _int } } = { 1 }
               \par\addvspace{ \l__enumext_minipage_after_skip }
           \end__enumext_mini_page
           \hfill
           \__enumext_mini_page{ \dim_use:c { l__enumext_minipage_right_ \__enumext_level: _dim } }
             \par\nointerlineskip
             \addvspace { \l__enumext_minipage_right_skip }
             \bool_if:nF {#1}
               {
1554
                 \centering
             \int_gzero:N \g__enumext_minipage_stat_int
```

58 / 156

(End of definition for $_$ enumext_mini_right_cmd:n.)

__enumext_keyans_mini_right_cmd:n

The function __enumext_keyans_mini_right_cmd:n takes as argument the *starred* '*' of the \miniright command in the keyans environment. The implementation of this function is the same as that of the __enumext_mini_right_cmd:n function of the enumext environment.

```
\cs_new_protected:Npn \__enumext_keyans_mini_right_cmd:n #1
1567
       \dim_compare:nNnTF { \l__enumext_minipage_right_v_dim } > { \c_zero_dim }
1568
1569
           \__enumext_keyans_multicols_stop:
           \int_compare:nNnT { \l__enumext_columns_v_int } = { 1 }
             {
                \par\addvspace{ \l__enumext_minipage_after_skip }
             }
1574
           \end__enumext_mini_page
           \hfill
           \__enumext_mini_page{ \l__enumext_minipage_right_v_dim }
             \par\nointerlineskip
             \addvspace { \l__enumext_minipage_right_skip }
             \bool_if:nF {#1}
1580
               {
1581
                  \centering
1582
1583
             \int_gzero:N \g__enumext_minipage_stat_int
1584
         { \msg_error:nnn { enumext } { wrong-miniright-use } }
1580
       % paranoia
       \RenewDocumentCommand \miniright { s }
1588
1589
           \msg_error:nn { enumext } { many-miniright-used }
1590
         }
1591
     }
1592
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_enumext_keyans_mini_right_cmd:n.)$

13.23 Setting above and below keys

While having controlled the *vertical spaces* within the enumext and keyans environments when using the columns or mini-env keys, sometimes the "*vertical spaces above*" or "*vertical spaces below*" the environments are not as expected and it is necessary to be able to apply a "*fine correction*" to these. As I have not been able to correct these *glitches*, the best option is to leave a couple of $\langle keys \rangle$ dedicated to this purpose, in this case it is best to use \vspace or \vspace* when convenient.

Define above, above*, below and below* keys for enumext and keyans environments.

```
ahove*
        \cs_set_protected:Npn \__enumext_tmp:nn #1 #2
below
        1594
             {
below*
               \keys_define:nn { enumext / #1 }
        1595
                 {
        1596
                          .skip_set:c = { l__enumext_vspace_above_#2_skip },
        1597
                         .value_required:n = true,
                   above
        1598
                   above* .code:n
                                     = \bool_set_true:c { l__enumext_vspace_a_star_#2_bool }
        1599
                                        \keys_set:nn { enumext / #1 } { above = {##1} },
                   above* .value_required:n = true,
                         .skip_set:c = { l__enumext_vspace_below_#2_skip },
                   below
        1603
                   below .value_required:n = true,
                   below* .code:n
                                      = \bool_set_true:c { l__enumext_vspace_b_star_#2_bool }
        1604
                                        \keys_set:nn { enumext / #1 } { below = {##1} },
        1605
                   below* .value_required:n = true,
        1606
                 }
        1607
        \(\text{\\_enumext_tmp:nn #1}\)
```

(End of definition for above and others.)

above

13.23.1 Functions for above and below keys in enumext

__enumext_vspace_above:

The function __enumext_vspace_above: apply the *vertical space above* the enumext environment set by the above* and above keys.

```
\cs_new_protected:Nn \__enumext_vspace_above:
    {
1611
       \skip_if_eq:nnF
1612
         { \skip_use:c { l__enumext_vspace_above_ \__enumext_level: _skip } } { \c_zero_skip }
1613
1614
           \bool_if:cTF { l__enumext_vspace_a_star_ \__enumext_level: _bool }
1615
               \vspace*{ \skip_use:c { l__enumext_vspace_above_ \__enumext_level: _skip } }
             }
               \vspace { \skip_use:c { l__enumext_vspace_above_ \__enumext_level: _skip } }
1621
         }
1622
1623
```

(End of definition for $_$ enumext_vspace_above:.)

__enumext_vspace_below:

The function __enumext_vspace_below: apply the *vertical space below* the enumext environment set by the below* and below keys.

 $(\textit{End of definition for } \verb|\|_enumext_vspace_below:.)$

13.23.2 Functions for above and below keys in keyans

__enumext_vspace_above_v:

The function __enumext_vspace_above_v: apply the *vertical space above* the keyans environment set by the above and above* keys.

(End of definition for __enumext_vspace_above_v:.)

__enumext_vspace_below_v:

The function __enumext_vspace_below_v: apply the *vertical space below* the keyans environment set by the below* and below keys.

(End of definition for $\c enumert vspace_below_v:.$)

13.23.3 Functions for above and below keys in enumext* keyans*

__enumext_vspace_above_vii:
 __enumext_vspace_above_viii:

The functions __enumext_vspace_above_vii: and __enumext_vspace_above_viii: apply the *vertical space above* the enumext* and keyans* environments set by the above and above* keys.

```
\cs_new_protected:Nn \__enumext_vspace_above_vii:
1661
       \skip_if_eq:nnF { \l__enumext_vspace_above_vii_skip } { \c_zero_skip }
1662
1663
            \bool_if:NTF \l__enumext_vspace_a_star_vii_bool
1664
             {
1665
                \vspace*{ \l__enumext_vspace_above_vii_skip }
             { \vspace { \l__enumext_vspace_above_vii_skip } }
   \cs_new_protected:Nn \__enumext_vspace_above_viii:
1672
       \skip_if_eq:nnF { \l__enumext_vspace_above_viii_skip } { \c_zero_skip }
1673
1674
           \bool_if:NTF \l__enumext_vspace_a_star_viii_bool
1675
1676
                \vspace*{ \l__enumext_vspace_above_viii_skip }
1677
1678
             { \vspace { \l__enumext_vspace_above_viii_skip } }
         }
1681
```

(End of definition for __enumext_vspace_above_vii: and __enumext_vspace_above_viii:.)

 The functions __enumext_vspace_below_vii: and __enumext_vspace_below_viii: apply the *vertical space below* the enumext* and keyans* environments set by the below* and below keys.

```
1682 \cs_new_protected:Nn \__enumext_vspace_below_vii:
       \skip_if_eq:nnF { \l__enumext_vspace_below_vii_skip } { \c_zero_skip }
           \bool_if:NTF \l__enumext_vspace_b_star_vii_bool
               \vspace*{ \l__enumext_vspace_below_vii_skip }
             { \vspace { \l__enumext_vspace_below_vii_skip } }
1691
   \cs_new_protected:Nn \__enumext_vspace_below_viii:
1693
       \skip_if_eq:nnF { \l__enumext_vspace_below_viii_skip } { \c_zero_skip }
           \bool_if:NTF \l__enumext_vspace_b_star_viii_bool
1698
               \vspace*{ \l__enumext_vspace_below_viii_skip }
1600
1700
             { \vspace { \l__enumext_vspace_below_viii_skip } }
1701
         }
     }
1703
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_vspace_below_vii:\ and\ \verb|_-enumext_vspace_below_viii:)$

13.24 Setting series, resume and resume* keys

The series key is responsible for the whole process of the resume and resume* keys. The idea behind this is to be able to absorb the $\langle keys \rangle$ passed to the *optional argument* of the "first level" of the environments enumext and enumext*, but, discarding some specific $\langle keys \rangle$. This implementation is adapted directly from the code provided by Jonathan P. Spratte (@Skillmon) in chat-TeX-SX

61/156

(End of definition for series, resume, and resume*.)

13.24.1 Internal functions for series key

__enumext_filter_series:n
 __enumext_filter_series_key:n
 __enumext_filter_series_pair:nn

The function __enumext_filter_series:n will be in charge of filtering the $\langle keys \rangle$ we want to store where $\{\#1\}$ represents the *optional argument* passed to the environment.

The function __enumext_filter_series_key:n will be responsible for filtering the $\langle keys \rangle$ that are passed "without value" by excluding the resume, resume* and base-fix keys.

The function $_$ _enumext_filter_series_pair:nn will be responsible for filtering the $\langle keys \rangle$ that are passed "with value" by excluding the series, resume, start, start*, save-ans and save-key keys.

 $(End\ of\ definition\ for\ _enumext_filter_series:n\ ,\ _enumext_filter_series_key:n\ ,\ and\ \setminus_enumext_filter_series_pair:nn.)$

__enumext_parse_series:n
__enumext_resume_last:n

The function __enumext_parse_series:n will be responsible for storing the filtered $\langle keys \rangle$ in the global variable \g__enumext_series_ $\langle series\ name \rangle$ _tl along with the creation of the integer variable \g__enumext_series_ $\langle series\ name \rangle$ _int when the key is passed as an argument; otherwise, it will check the state of the boolean variable \l_enumext_resume_active_bool set by the keys resume and resume* and will call the function _enumext_resume_last:n.

The value of boolean variable \l__enumext_resume_active_bool is set to true by the function __enumext_resume_counter:n which is used by the keys resume and resume*, in this case we must Make sure it is set to false so that it does not overwrite the default filtered \(\lambda eys \rangle \). This function is passed to the function __enumext_parse_keys:n in the enumext environment definition (\subseteq 13.38) and to the function __enumext_parse_keys_vii:n in the enumext* environment definition (\subseteq 13.43).

```
1742 \cs_new_protected:Npn \__enumext_parse_series:n #1
1743 {
1744 \str_if_empty:NTF \l__enumext_series_str
1745 {
1746 \bool_if:NF \l__enumext_resume_active_bool
1747 {
1748 \__enumext_resume_last:n {#1}
1749 }
1750 }
1751 {
1752 \tag{tl_gclear_new:c { g__enumext_series_ \l__enumext_series_str_tl }
1753 \tag{tl_gset:ce { g__enumext_series_ \l__enumext_series_str_tl }
1754 { \__enumext_filter_series:n {#1} }
1755 \tag{nt_if_exist:cF { g__enumext_series_ \l__enumext_series_str_int }
1756 {
1756 }
1756 }
1757 \tag{nt_if_exist:cF { g__enumext_series_ \l__enumext_series_str_int }
1758 }
1759 \tag{nt_if_exist:cF { g__enumext_series_ \l__enumext_series_str_int }
1750 }
1750 \tag{nt_if_exist:cF { g__enumext_series_ \l__enumext_series_str_int }
1751 }
1752 \tag{nt_if_exist:cF { g__enumext_series_ \l__enumext_series_str_int }
1753 }
1754 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_str_int }
1755 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_str_int }
1756 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_str_int }
1758 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_str_int }
1759 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_ \l_enumext_series_str_int }
1759 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_ \l_enumext_series_ \l_enumext_series_str_int }
1759 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_ \l_enumext_series_ \l_enumext_series_ \l_enumext_series_ \l_enumext_series_str_int }
1750 \tag{nt_if_exist:cF { g__enumext_series_ \l_enumext_series_ \l
```

The function __enumext_resume_last:n will be in charge of saving the filtering $\langle keys \rangle$ when the series key is *not used* and will save them in the variable \g__enumext_standar_series_tl for the enumext environment and in the variable \g_enumext_starred_series_tl for the enumext* environment.

(End of definition for __enumext_parse_series:n and __enumext_resume_last:n.)

13.24.2 Internal function to save counter value

__enumext_resume_save_counter:

The __enumext_resume_save_counter: function will save the last counter value to \g__enumext_-series_ $\langle series\ name \rangle$ _int if the series= $\{\langle series\ name \rangle\}$ key has been passed, to \g__enumext_resume_-int if it has passed the key resume without value and the key series is not active, in \g__enumext_series_- $\langle series\ name \rangle$ _int if the key resume= $\{\langle series\ name \rangle\}$ has been passed and in \g__enumext_series_ $\langle store\ name \rangle$ _int if the key has been passed save-ans= $\{\langle store\ name \rangle\}$.

The variables \l__enumext_series_str and \l__enumext__resume_name_tl contain the same $\{\langle series \ name \rangle\}$ but are executed at different moments, the integer variable with \l__enumext_series_str sets the value when execute series= $\{\langle series \ name \rangle\}$ and the integer variable with \l__enumext__resume_name_tl sets the subsequent values when use resume= $\{\langle series \ name \rangle\}$. This function is passed to the enumext environment definition (§13.38) and the enumext* environment definition (§13.43).

```
\cs_new_protected:Nn \__enumext_resume_save_counter:
       \bool_if:NT \g__enumext_standar_bool
1776
           \tl_if_empty:NF \l__enumext_series_str
1778
               \int_gset_eq:cN
                 { g__enumext_series_ \l__enumext_series_str _int } \value{enumXi}
178
           \tl_if_empty:NTF \l__enumext_resume_name_tl
               \str_if_empty:NT \l__enumext_series_str
                    \int_gset_eq:NN \g__enumext_resume_int \value{enumXi}
1788
             }
               \int_if_exist:cT { g__enumext_series_ \l__enumext_resume_name_tl _int }
                   \int_gset_eq:cN
                      { g__enumext_series_ \l__enumext_resume_name_tl _int } \value{enumXi}
1796
             }
           \int_if_exist:cT { g__enumext_resume_ \l__enumext_store_name_tl _int }
1797
             {
1798
               \int_gset_eq:cN
1799
                 { g__enumext_resume_ \l__enumext_store_name_tl _int } \value{enumXi}
       \bool_if:NT \g__enumext_starred_bool
           \tl_if_empty:NF \l__enumext_series_str
1805
1806
             {
1807
               \int_gset_eq:cN
                 { g__enumext_series_ \l__enumext_series_str _int } \value{enumXvii}
1808
```

```
1810
            \tl_if_empty:NTF \l__enumext_resume_name_tl
1811
                \str_if_empty:NT \l__enumext_series_str
1812
1813
                     \int_gset_eq:NN \g__enumext_resume_vii_int \value{enumXvii}
1814
1815
              }
                \int_if_exist:cT { g__enumext_series_ \l__enumext_resume_name_tl _int }
                    \int_gset_eq:cN
                       { g__enumext_series_ \l__enumext_resume_name_tl _int } \value{enumXvii}
1821
                  }
1822
              }
1823
            \int_if_exist:cT { g__enumext_resume_ \l__enumext_store_name_tl _int }
1824
              {
1825
                \int_gset_eq:cN
1826
                  { g__enumext_resume_ \l__enumext_store_name_tl _int } \value{enumXvii}
1827
         }
     }
```

(End of definition for __enumext_resume_save_counter:.)

13.24.3 Internal functions for resume key

__enumext_resume_series:n

The function __enumext_resume_series:n will handle the argument passed to the resume key in enumext and enumext* environments. If the key is passed without value the function __enumext_resume_counter: is executed which will set the counter according to the numbering of the last enumext or enumext* environments in which series={\series name\ranger} key is not present, if the save-ans key is active it will set the counter according to the value of the integer variable created by that key, otherwise it will verify that the \g__enumext_series_\series name\ranger_tl variable set by the series key exists, if so it will pass these keys to the first level of the environment, otherwise it will return an error.

```
\cs_new_protected:Npn \__enumext_resume_series:n #1
     {
1832
       \tl_if_empty:nTF {#1}
1833
         {
1834
              _enumext_resume_counter:n { }
1835
         }
         {
            \tl_if_exist:cTF { g__enumext_series_ \tl_to_str:n {#1} _tl }
1838
              {
                   enumext resume counter:n {#1}
1840
                \bool_if:NT \g__enumext_standar_bool
1841
                  {
1842
                     \keys_set:nv { enumext / level-1 }
1843
                       { g__enumext_series_ \tl_to_str:n {#1} _tl }
1844
                  }
                \bool_if:NT \g__enumext_starred_bool
                  {
                    \keys_set:nv { enumext / enumext* }
                       { g__enumext_series_ \tl_to_str:n {#1} _tl }
              }
1851
                \bool_if:NT \g__enumext_standar_bool
1853
1854
                    \msg_error:nnn { enumext } { unknown-series } {#1}
                  }
                \bool_if:NT \g__enumext_starred_bool
                  {
                    \msg_error:nnn { enumext } { unknown-series } {#1}
                  }
              }
1861
         }
1862
1863
```

(End of definition for __enumext_resume_series:n.)

```
__enumext_resume_counter:n
\__enumext_resume_counter:
  \__enumext_resume_counter_series:
```

The function __enumext_resume_counter:n will set the variable \l__enumext_resume_active_bool to true and pass the value of the key resume to the variable \l__enumext_series_name_tl which will @2024 by Pablo González L

64/156

contain the $\{\langle series\ name \rangle\}$. If the variable \l__enumext_series_name_tl is empty, that is, we are passing the key resume without value, we will execute the function __enumext_resume_counter: otherwise, when we pass resume= $\{\langle series\ name \rangle\}$ we will execute the function __enumext_resume_counter_series:, finally we will execute the function __enumext_resume_counter_save_ans: which is associated with the key save-ans.

```
\cs_new_protected:Npn \__enumext_resume_counter:n #1
1865
    {
       \bool_set_true:N \l__enumext_resume_active_bool
1866
       \tl_set:Nn \l__enumext_resume_name_tl {#1}
1867
       \tl_if_empty:NTF \l__enumext_resume_name_tl
1868
              _enumext_resume_counter:
1870
         }
1871
         {
1872
              _enumext_resume_counter_series:
1873
         }
         _enumext_resume_counter_save_ans:
1876
```

The __enumext_resume_counter: function is executed when the resume key is used *without value*, only the counters for the "first level" of the environments will be set.

```
\cs_new_protected:Nn \__enumext_resume_counter:
1878
       \bool_if:NT \g__enumext_standar_bool
1879
           \int_gincr:N \g__enumext_resume_int
           \int_set_eq:NN \l__enumext_start_i_int \g__enumext_resume_int
       \bool_if:NT \g__enumext_starred_bool
1884
1885
         {
           \int_gincr:N \g__enumext_resume_vii_int
1886
           \int_set_eq:NN \l__enumext_start_vii_int \g__enumext_resume_vii_int
1887
1888
1889
```

The function __enumext_resume_counter_series: will be executed when the resume= $\{\langle series \ name \rangle\}$ key is active, setting the counters for the "first level" of the environments according to the value of the integer variables created by the series key.

The function __enumext_resume_counter_save_ans: will be executed when the save-ans key is active along with the resume key, setting the counters for the "first level" of the environments according to the value of the integer variables created by the save-ans key.

(End of definition for $_$ enumext_resume_counter:n and others.)

13.24.4 Internal function for resume* key

__enumext_resume_starred:

The function __enumext_resume_starred: will handle the resume* key in the enumext and enumext* environments. This function will execute the filtered $\langle keys \rangle$ in the last one and will continue with the numbering according to the last execution of the environment enumext or enumext* in which the keys resume= $\{\langle series name \rangle\}$ or series= $\{\langle series name \rangle\}$ were not active.

```
1928 \cs_new_protected:Nn \__enumext_resume_starred:
       \bool_if:NT \g_enumext\_standar\_bool
1930
1931
           \tl_if_empty:NF \g__enumext_standar_series_tl
1932
1933
                \__enumext_resume_counter:n { }
                \keys_set:nV { enumext / level-1 } \g__enumext_standar_series_tl
1935
1937
       \bool_if:NT \g__enumext_starred_bool
           \tl_if_empty:NF \g__enumext_starred_series_tl
                \__enumext_resume_counter:n { }
                \keys_set:nV { enumext / enumext* } \g__enumext_starred_series_tl
1943
1944
         }
1945
```

(End of definition for __enumext_resume_starred:.)

13.25 Setting save-ans, check-ans and no-store keys

The key save-ans is directly associated with the keys check-ans, no-store, resume and resume*, this will activate the entire "storage system" in the enumext package.

13.25.1 Setting save-ans key

save-ans We define the keys save-ans only for the "first level" of enumext and enumext*.

(End of definition for save-ans.)

13.25.2 Internal functions for save-ans key

__enumext_start_save_ans_msg:
\ enumext stop save ans msg:

The functions __enumext_start_save_ans_msg: and __enumext_stop_save_ans_msg: will display in the terminal and .log file the environment in which the save-ans key was executed along with the line at the beginning and end of it. The function __enumext_start_save_ans_msg: will be passed to __enumext_storing_set:n and the function __enumext_stop_save_ans_msg: will be passed to the function __enumext_execute_after_env:.

```
1956 \cs_new_protected:Nn \__enumext_start_save_ans_msg:
1957 {
1958    \msg_term:nnVV { enumext } { save-ans-log }
1959    \g__enumext_envir_name_tl \l__enumext_store_name_tl
1960 }
1961 \cs_new_protected:Nn \__enumext_stop_save_ans_msg:
1962 {
```

```
\msg_term:nnVV { enumext } { save-ans-log-hook }

\g__enumext_envir_name_tl \g__enumext_store_name_tl

\g_ { logs }

(End of definition for \__enumext_start_save_ans_msg: and \__enumext_stop_save_ans_msg:)
```

__enumext_storing_set:n
__enumext_storing_exec:

The function __enumext_storing_set:n first pass the value of the save-ans key to the variable \l__enumext_store_name_tl which will contain the $\{\langle store\ name \rangle\}$ of the sequence and prop list we will use. If \l__enumext_store_name_tl is empty we return an error message, otherwise will return the appropriate message __enumext_start_save_ans_msg: and proceed to execute the function __enumext_storing_exec: for enumext and enumext* environments.

```
1966 \cs_new_protected:Npn \__enumext_storing_set:n #1
1967
       \tl set:Ne \l enumext store name tl {#1}
1968
       \tl_if_empty:NTF \l__enumext_store_name_tl
1969
1970
           \bool_lazy_or:nnT
1971
             { \l__enumext_standar_first_bool } { \l__enumext_starred_first_bool }
1972
               \msg_error:nnV { enumext } { save-ans-empty } \g__enumext_envir_name_tl
         }
         {
           \bool_lazy_or:nnT
             { \l__enumext_standar_first_bool } { \l__enumext_starred_first_bool }
               \__enumext_start_save_ans_msg:
               \__enumext_storing_exec:
         }
```

The function __enumext_storing_exec: will set to true the variable \l__enumext_store_active_bool which activates the use of the \anskey command and the anskey*, keyans, keyans* and keyanspic environments and will set to "true" the variable \l__enumext_check_answers_bool used for intenal checking answers mechanism set by the check-ans and no-store keys, copy $\{\langle store\ name \rangle\}$ into the variable \g__enumext_store_name_tl and execute the function __enumext_anskey_env_make: V creating the environment anskey* ($\S13.30$).

```
1986 \cs_new_protected:Nn \__enumext_storing_exec:
1987 {
1988    \bool_set_true:N \l__enumext_store_active_bool
1989    \bool_set_true:N \l__enumext_check_answers_bool
1990    \tl_gset:NV \g__enumext_store_name_tl \l__enumext_store_name_tl
1991    \__enumext_anskey_env_make:V \l__enumext_store_name_tl
```

The prop list $\g_{\text{enumext_series}}\store\ name\$ _prop and the sequence $\g_{\text{enumext_series}}\store\ name\$ _seq will be created globally to "store content" in case they do not exist together with the integer variable $\g_{\text{enumext_series}}\store\ name\$ _int used by the keys resume and resume*.

```
\prop_if_exist:cF { g__enumext_ \l__enumext_store_name_tl _prop }
         {
           \msg_log:nnV { enumext } { store-prop } \l__enumext_store_name_tl
           \prop_new:c { g__enumext_ \l__enumext_store_name_tl _prop }
         }
1996
       \seq_if_exist:cF { g__enumext_ \l__enumext_store_name_tl _seq }
1997
1998
           \msg_log:nnV { enumext } { store-seq } \l__enumext_store_name_tl
           \seq_new:c { g__enumext_ \l__enumext_store_name_tl _seq }
       \int_if_exist:cF { g__enumext_resume_ \l__enumext_store_name_tl _int }
           \msg_log:nnV { enumext } { store-int } \l__enumext_store_name_tl
           \int_new:c { g__enumext_resume_ \l__enumext_store_name_tl _int }
2006
    }
2007
```

 $(\textit{End of definition for } \c\c\c) = \texttt{enumext_storing_set:n} \ \ \textit{and } \c\c\c\c)$

13.25.3 The check answer mechanism

The internal mechanism for "checking answers" follows this logic:

If the line begins with \item or \item* and does NOT open a nested environment, each \item or \item* must contain a single execution of the \anskey command, i.e. the counter of the executions of the \anskey command must be equal to the counter associated with the sum of executions of \item and \item*.

If the line begins with \item or \item* and opens a nested environment each \item or \item* in the nested environment must have a *single* execution of the \anskey command and the counter associated to the sum of \item and \item* executions must decrementing by "one" to maintain equality.

In order for the mechanism for the check-answer to work (not counting keyans, keyans* and keyanspic) we need:

- 1. We must keep track of the total number of \item and \item* (enumerated) that appear within the environment including the nested levels.
- 2. We must keep track of the total number of \item and \item* (enumerated) that appear per level of nesting.
- 3. Keeping track of the number of times the environment nests.

The integer variable associated to the sum of each $\idesign* in the environment <math>g_{enumext}-idem_number_int must match the integer variable <math>g_{enumext}-idem_num must match the integer variable gas variable gas va$

- a) If the list only has one level the number of \idet + \idet = \arrange anskey
- b) If the list has *nested levels*, for each level of nesting we need to decrementing by one (for the \item or \item* that opens the nest) so that the account remains the same.

With keyans, keyans* and keyanspic it is enough to increase in one the integer of \anskey. The integers created must be global if they are not lost in the interior levels of nesting and to execute the test we will use a "hook" function after closing the *first level* of the environment.

13.25.4 Setting check-ans and no-store keys

no-store

Now we define the keys check-ans and no-store for all levels of enumext and enumext* environments.

```
2008 \cs_set_protected:Npn \__enumext_tmp:n #1
     {
2009
       \keys_define:nn { enumext / #1 }
2010
         {
           check-ans .bool_set:N = \l__enumext_check_ans_key_bool,
           check-ans .initial:n = false,
2013
           check-ans .value_required:n = true,
2014
           no-store .code:n = {
2015
                                   \bool_set_false:N \l__enumext_check_answers_bool
2016
                                   \bool_set_false:N \l__enumext_check_ans_key_bool
2017
                                 },
2018
                     .value_forbidden:n = true,
2019
         }
2022 \clist_map_inline:nn
    {
       level-1, level-2, level-3, level-4, enumext*
2024
     }
2025
     { \__enumext_tmp:n {#1} }
```

 $(\mathit{End}\ of\ definition\ for\ check-ans\ \ and\ no\text{-store.})$

13.25.5 Set-up check answer mechanism

__enumext_check_ans_active:
__enumext_check_ans_level:

The function __enumext_check_ans_active: will first check the state of the variable \l__enumext_store_name_tl, that is, the save-ans key is active, if so it will check the state of the variable \l__enumext_check_answers_bool handled by the key no-store and will execute the function __enumext_check_ans_level: only if "true", i.e. the key no-store is not active.

The function __enumext_check_ans_level: will decrement by "one" the value of the variable \g__-enumext_item_number_int which keeps track of the executions of \item and \item* for each level of nesting of the environment enumext, taking into account whether it is nested within enumext* or the opposite and set \l__enumext_item_number_bool to "false".

```
2037 \cs_new_protected:Nn \__enumext_check_ans_level:
    {
2038
       \int_case:nn { \l__enumext_level_int }
2040
           { 1 }{
2041
                  \bool_lazy_all:nT
                      { \bool_if_p:N \g__enumext_starred_bool }
                      { \int_compare_p:nNn { \l__enumext_level_h_int } = { 1 } }
                      \int_gdecr:N \g__enumext_item_number_int
                      \bool_set_false:N \l__enumext_item_number_bool
           { 2 }{
                  \int_gdecr:N \g__enumext_item_number_int
                  \bool_set_false:N \l__enumext_item_number_bool
           { 3 }{
                  \int_gdecr:N \g__enumext_item_number_int
                  \bool_set_false:N \l__enumext_item_number_bool
           { 4 }{
                  \int_gdecr:N \g__enumext_item_number_int
                  \bool_set_false:N \l__enumext_item_number_bool
```

We should only execute this if enumext* is nested in the "first level" of enumext, for the rest of the cases the value of \g_enumext_item_number_int is already decreased.

```
\int_case:nn { \l__enumext_level_h_int }
         {
2066
           { 1 }{
2067
                   \bool_lazy_all:nT
                     {
                       { \bool_if_p:N \g__enumext_standar_bool }
                       { \int_compare_p:nNn { \l__enumext_level_int } = { 1 } }
                       \int_gdecr:N \g__enumext_item_number_int
                       \bool_set_false:N \l__enumext_item_number_bool
2076
                 }
2077
         }
2078
2079
```

 $(\textit{End of definition for } \verb|_enumext_check_ans_active: and \verb|_enumext_check_ans_level:|)$

__enumext_check_ans_key_hook:

The function __enumext_check_ans_key_hook: will *export* the status of the local variable \l__enumext_check_ans_key_bool to the global variable \g__enumext_check_ans_key_bool only if the key check-ans is active.

```
2080 \cs_new_protected:Nn \__enumext_check_ans_key_hook:
2081
    {
       \bool_lazy_and:nnT
2082
         { \bool_if_p:N \l__enumext_check_ans_key_bool }
2083
         { \bool_if_p:N \g__enumext_standar_bool }
2084
         {
           \bool_gset_true:N \g__enumext_check_ans_key_bool
       \bool_lazy_and:nnT
         { \bool_if_p:N \l__enumext_check_ans_key_bool }
         { \bool_if_p:N \g__enumext_starred_bool }
           \bool_gset_true:N \g__enumext_check_ans_key_bool
    }
2094
```

(End of definition for $_=$ enumext_check_ans_key_hook:.)

__enumext_item_answer_diff:

The function __enumext_item_answer_diff: will set the value of the variable \g__enumext_item_-answer_diff_int which is used by the functions __enumext_check_ans_show: for the key save-ans and by the function __enumext_check_ans_log: by the internal "check answer" mechanism. This function will be passed to the function __enumext_execute_after_env:.

(End of definition for __enumext_item_answer_diff:.)

__enumext_check_ans_msg_less:
_enumext_check_ans_msg_same_ok:
_enumext_check_ans_msg_greater:

The function __enumext_check_ans_show: will be executed within the function __enumext_execute_-after_env: when the key check-ans is active, that is, when \g__enumext_check_ans_key_bool is "true" and will return the appropriate message according to the value of \g__enumext_item_answer_diff_int set by the function __enumext_item_answer_diff:.

```
\cs_new_protected:Nn \__enumext_check_ans_show:
2103
       \int_case:nn { \g__enumext_item_answer_diff_int }
2104
           { -1 }{ \__enumext_check_ans_msg_less:
             0 }{ \__enumext_check_ans_msg_same_ok: }
             1 }{ \__enumext_check_ans_msg_greater: }
   \cs_new_protected:Nn \__enumext_check_ans_msg_less:
       \msg_warning:nneee { enumext } { item-less-answer } { \g_enumext_store_name_tl }
         { \g__enumext_envir_name_tl } { \g__enumext_start_line_tl }
   \cs_new_protected:Nn \__enumext_check_ans_msg_same_ok:
       \msg_term:nneee { enumext } { items-same-answer } { \g_enumext_store_name_tl }
         { \g__enumext_envir_name_tl } { \g__enumext_start_line_tl }
   \cs_new_protected:Nn \__enumext_check_ans_msg_greater:
       \msg_warning:nneee { enumext } { item-greater-answer } { \g__enumext_store_name_tl }
         { \g__enumext_envir_name_tl } { \g__enumext_start_line_tl }
2124
```

(End of definition for $\ensuremath{\backslash}$ enumext_check_ans_show: and others.)

__enumext_check_ans_log_msg_less:
_enumext_check_ans_log_msg_same_ok:
_enumext_check_ans_log_msg_greater:

The function __enumext_check_ans_log: will be executed within the function __enumext_execute_-after_env: when the key check-ans is not active, that is, when \g__enumext_check_ans_key_bool is "false" and write in the log the appropriate message according to the value of \g__enumext_item_answer_-diff_int set by the function __enumext_item_answer_diff:.

```
\cs_new_protected:Nn \__enumext_check_ans_log:
       \int_case:nn { \g__enumext_item_answer_diff_int }
2128
         {
           { -1 }{ \__enumext_check_ans_log_msg_less:
             0 }{ \__enumext_check_ans_log_msg_same_ok: }
           { 1 }{ \__enumext_check_ans_log_msg_greater: }
   \cs_new_protected:Nn \__enumext_check_ans_log_msg_less:
2136
       \msg_log:nneee { enumext } { item-less-answer } { \g__enumext_store_name_tl }
         { \g__enumext_envir_name_tl } { \g__enumext_start_line_tl }
   \cs_new_protected:Nn \__enumext_check_ans_log_msg_same_ok:
       \msg_log:nneee { enumext } { items-same-answer } { \g__enumext_store_name_tl }
2142
         { \g__enumext_envir_name_tl } { \g__enumext_start_line_tl }
2143
2144
```

(End of definition for __enumext_check_ans_log: and others.)

13.25.6 Check for \item* and \anspic* commands

\ enumext check starred cmd:n

The function __enumext_check_starred_cmd:n performs an *extra check* for the keyans, keyans* and keyanspic environments. Unlike the *check* executed by check-ans key this one is not controlled by any key, it is intended to prevent the forgetting of \item* or \anspic* in these environments.

```
2150 \cs_new_protected:Npn \__enumext_check_starred_cmd:n #1
2152
       \int_compare:nNnT
         { \g__enumext_check_starred_cmd_int } = { 0 }
2154
           \msg_warning:nnnV
             { enumext } { missing-starred }{ #1 } \l__enumext_check_start_line_env_tl
         }
       \int compare:nNnT
2158
         { \g__enumext_check_starred_cmd_int } > { 1 }
           \msg_warning:nnnV
              { enumext } { many-starred }{ #1 } \l__enumext_check_start_line_env_tl
2164
       \int_gzero:N \g__enumext_check_starred_cmd_int
       \tl_clear:N \l__enumext_check_start_line_env_tl
2165
2166
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_check_starred_cmd:n.)$

13.26 Keys and functions associated with storage

We add the keys wrap-ans, wrap-opt, save-sep, mark-ans, mark-pos, show-ans, show-pos, mark-ref wrap-ans and save-ref related to the "storage system" and internal mechanism of "label and ref" only at the first level wrap-opt of enumext and enumext*. save-sep mark-ans 2167 \cs_set_protected:Npn __enumext_tmp:n #1 mark-pos show-ans \keys_define:nn { enumext / #1 } 2169 mark-ref wrap-ans .cs_set_protected:Np = __enumext_anskey_wrapper:n ##1, save-ref wrap-ans .initial:n = \fbox{\parbox[t]{\dimeval{\itemwidth -2\fboxsep -2\fboxrule}}{##1}} }, .value_required:n = true, wrap-ans 2176 wrap-opt .cs_set_protected:Np = __enumext_keyans_wrapper_opt:n ##1, wrap-opt .initial:n = [{##1}], 2178 .value_required:n = true, wrap-opt .tl_set:N = \l__enumext_store_keyans_item_opt_sep_tl, save-sep save-sep .initial:n = {, ~ }, save-sep .value_required:n = true, mark-ans .tl_set:N = \l__enumext_mark_answer_sym_tl, .initial:n = \textasteriskcentered, mark-ans 2184 .value required:n = true. 2185 mark-ans .choice:, 2186 mark-pos mark-pos / left .code:n = \str_set:Nn \l__enumext_mark_position_str { l }, 2187 mark-pos / right .code:n = \str_set:Nn \l__enumext_mark_position_str { r }, 2188 mark-pos / unknown .code:n = 2189 \msg_error:nneee { enumext } { unknown-choice } 2190 { mark-pos } { left, ~ right } { \exp_not:n {##1} }, mark-pos .initial:n = right, mark-pos .value_required:n = true, .bool_set:N = \l__enumext_show_answer_bool, show-ans .initial:n = false, show-ans show-ans .value required:n = true. 2196 .bool_set:N = \l__enumext_show_position_bool, show-pos show-pos .initial:n = false, 2198 show-pos .value_required:n = true, 2199 mark-ref .tl_set:N = \l__enumext_mark_ref_sym_tl,

```
.initial:n = \textreferencemark,
                      mark-ref
                      mark-ref
                                  .value_required:n = true,
                                 .bool_set:N = \l__enumext_store_ref_key_bool,
           2203
                      save-ref
                                 .initial:n = false,
                      save-ref
          2204
                                 .value_required:n = true,
                      save-ref
          2205
          2206
          2207
          2208 \clist_map_inline:nn { level-1, enumext* } { \__enumext_tmp:n {#1} }
          (End of definition for wrap-ans and others.)
          For the keyans and keyans* environments we will only add the keys mark-pos, show-ans and show-pos.
show-ans
           \cs_set_protected:Npn \__enumext_tmp:n #1
show-pos
          2210
                  \keys_define:nn { enumext / #1 }
          2211
                    {
                      mark-pos .choice:,
                      mark-pos / left .code:n = \str_set:Nn \l__enumext_mark_position_str { l },
                      mark-pos / right .code:n = \str_set:Nn \l__enumext_mark_position_str { r },
                      mark-pos .initial:n = right,
           2216
                      mark-pos .value_required:n = true,
          2217
                      show-ans .bool_set:N = \l__enumext_show_answer_bool,
                      show-ans .initial:n = false,
          2219
                      show-ans .value_required:n = true,
                      show-pos .bool_set:N = \l__enumext_show_position_bool,
                      show-pos .initial:n = false,
                      show-pos .value_required:n = true,
          2226 \clist_map_inline:nn { keyans, keyans* } { \__enumext_tmp:n {#1} }
```

(End of definition for mark-pos, show-ans, and show-pos.)

13.26.1 Storing structure of the environments

The idea behind "storing structure" in the sequence is to have a copy of the structure of the environment in which the key save-ans is being executed so we must capture the optional argument passed to the levels of the environment in which it is executed and "storing" this in the sequence.

__enumext_store_active_keys:n __enumext_store_active_keys_vii:n The functions __enumext_store_active_keys:n and __enumext_store_active_keys_vii:n will be responsible for the "storing keys" filtered from the optional argument of the environment in which the key save-ans is executed and the levels within this for the enumext and enumext* environments. We will execute this function only if the variable \l__enumext_store_save_key_X_bool is false, that is, the key store-key is not active, establishing the variable \l__enumext_store_save_key_X_tl with the filtered $\langle keys \rangle$.

```
2227 \cs_new_protected:Npn \__enumext_store_active_keys:n #1
       \bool_if:cF { l__enumext_store_save_key_ \__enumext_level: _bool }
           \tl_clear:c { l__enumext_save_key_ \__enumext_level: _tl }
2231
           \tl set:ce
             { l__enumext_store_save_key_ \__enumext_level: _tl }
             { \__enumext_filter_save_key:n {#1} }
2236
   \cs_new_protected:Npn \__enumext_store_active_keys_vii:n #1
2238
       \bool_if:NF \l__enumext_store_save_key_vii_bool
2239
         {
2240
           \tl_clear:N \l__enumext_store_save_key_vii_tl
2241
           \tl_set:Ne \l__enumext_store_save_key_vii_tl { \__enumext_filter_save_key:n {#1} }
2242
2243
     }
2244
```

 $(\textit{End of definition for } \verb|\|_enumext_store_active_keys:n | and \verb|\|_enumext_store_active_keys_vii:n.)|$

13.26.2 Setting save-key key

Since this "storing structure" in the sequence established by the save-ans key when executing \anskey or anskey*, we will not be able to modify it. The best thing here is to have a key that allows you to modify the optional argument of the "storing structure" in the sequence.

save-key

The values set by this key passed in the *optional argument* of the enumext and enumext* environments will override the values of the \l_enumext_store_save_key_X_tl variable set by the functions _enumext_store_active_keys:n and _enumext_store_active_keys_vii:n. Now define the key save-key for all levels of enumext and enumext* environments.

(End of definition for save-key.)

__enumext_parse_save_key:n
_enumext_parse_save_key_vii:n

The functions __enumext_parse_save_key:n and __enumext_parse_save_key_vii:n will be responsible for "storing keys" in the variable \l__enumext_store_save_key_X_tl for enumext and enumext*.

 $(\textit{End of definition for } \verb|\|_enumext_parse_save_key:n | and \verb|\|_enumext_parse_save_key_vii:n.)$

13.26.3 Internal functions to store optional arguments

__enumext_filter_save_key:n
 __enumext_filter_save_key_key:n
 __enumext_filter_save_key_pair:nn

The function __enumext_filter_save_key:n will be in charge of "filtering keys" we want to stored in sequence where {#1} represents the optional argument passed to the environment.

The function __enumext_filter_save_key_key:n will be responsible for "filtering keys" that are passed "without value" by excluding the resume, resume*, no-store and base-fix keys.

©2024 by Pablo González L 73 / 156

The function __enumext_filter_save_key_pair:nn will be responsible for "filtering keys" that are passed *"with value"* by excluding the series, resume, save-ans, save-ref, check-ans, show-ans, save-pos, wrap-ans, mark-ans, wrap-opt, save-sep, mark-ref, mini-env, mini-sep, mini-right and miniright* keys.

```
\cs_new:Npn \__enumext_filter_save_key_pair:nn #1#2
    {
2291
       \str_case:nnF {#1}
2292
         {
2293
           { series } {} { resume
                                       } {} { save-ans } {} { save-ref
2294
           { save-key } {} { check-ans } {} { show-ans } {} { show-pos } {}
           { wrap-ans } {} { mark-ans } {} { wrap-opt } {} { save-sep } {}
           { mark-ref } {} { mini-env } {} { mini-sep } {} { mini-right } {}
           { mini-right* } {}
         }
         { , { \exp_not:n {\#1}} } = { \exp_not:n {\#2} } }
2301
```

save_key_pair:nn.)

13.26.4 Function for storing content in prop list

__enumext_store_addto_prop:n __enumext_store_addto_prop:V The function $\ \ \$ enumext_store_addto_prop:n stores the $\{\langle content \rangle\}$ in prop list defined by save-ans key. The "stored content" is retrieved by means of the \getkeyans command.

The form in which the $\{\langle content \rangle\}$ is "stored" in the prop list is $\{\langle position \rangle\} \{\langle content \rangle\}$. This function is used by \anskey in enumext and enumext* environments, \item* in keyans and keyans* environments and \anspic* in keyanspic environment.

```
2302 \cs_new_protected:Npn \__enumext_store_addto_prop:n #1
2303
       \prop_gput_if_not_in:cen { g__enumext_ \l__enumext_store_name_tl _prop }
2304
2305
            \int_eval:n { \prop_count:c { g__enumext_ \l__enumext_store_name_tl _prop } + 1 }
2306
         }
2307
         { #1 }
2308
2309
2310 \cs_generate_variant:Nn \__enumext_store_addto_prop:n { V }
```

(End of definition for $_$ enumext_store_addto_prop:n.)

13.26.5 Function for storing content in sequence

The function $_$ enumext_store_addto_seq:n stores the $\{\langle content \rangle\}$ in sequence defined by save-ans key. This function is used by \anskey in enumext, \item* in keyans and \anspic in keyanspic.

The form in which the $\{\langle content \rangle\}$ is stored in sequence is in a internal enumext or enumext* environments with the "same structure" in which the command was executed.

The "stored content" is retrieved by means of the \printkeyans command.

```
2311 \cs_new_protected:Npn \__enumext_store_addto_seq:n #1
     {
2312
       \seq_gput_right:cn { g__enumext_ \l__enumext_store_name_tl _seq } { #1 }
     }
2315 \cs_generate_variant:Nn \__enumext_store_addto_seq:n { v, V }
```

 $(\mathit{End of definition for} \setminus __enumext_store_addto_seq:n.)$

13.26.6 Functions for storing structure in the sequence

__enumext_store_level_open: \ enumext store level close: The "storing structure" is handled by the functions __enumext_store_level_open: and __enumext_store_level_close: which are executed per level within the enumext environment.

```
\cs_new_protected:Nn \__enumext_store_level_open:
       \bool_if:NT \l__enumext_check_answers_bool
2318
           \tl_if_empty:cTF { l__enumext_store_save_key_ \__enumext_level: _tl }
                  _enumext_store_addto_seq:n
                 {
                   \item \begin{enumext}
             }
               \tl_put_left:cn { l__enumext_store_save_key_ \__enumext_level: _tl }
```

__enumext_store_addto_seq:n __enumext_store_addto_seq:v __enumext_store_addto_seq:V

```
\item \begin{enumext} [
                 \tl_put_right:cn { l__enumext_store_save_key_ \__enumext_level: _tl }
                   }
                 \__enumext_store_addto_seq:v { l__enumext_store_save_key_ \__enumext_level: _tl }
          }
     }
   \cs_new_protected:Nn \__enumext_store_level_close:
        \bool_if:NT \l__enumext_check_answers_bool
          {
               _enumext_store_addto_seq:n { \end{enumext} }
2345
2346
(\textit{End of definition for } \verb|\_enumext_store_level_open: and \verb|\_enumext_store_level_close:|)
The "storing structure" is handled by the functions \__enumext_store_level_open_vii: and \__enumext_-
store_level_close_vii: which are executed in the enumext* environment.
   \cs_new_protected:Nn \__enumext_store_level_open_vii:
2348
        \bool_if:NT \l__enumext_check_answers_bool
2349
            \tl_if_empty:NTF \l__enumext_store_save_key_vii_tl
              {
                    _enumext_store_addto_seq:n
                   {
                     \item \begin{enumext*}
                   }
              }
2358
                 \tl_put_left:Nn \l__enumext_store_save_key_vii_tl
                   {
                     \item \begin{enumext*}[
                 \tl_put_right:Nn \l__enumext_store_save_key_vii_tl
                   {
2366
                   _enumext_store_addto_seq:V \l__enumext_store_save_key_vii_tl
2367
2368
          }
2369
2370
   \cs_new_protected:Nn \__enumext_store_level_close_vii:
2371
2372
        \bool_if:NT \l__enumext_check_answers_bool
               _enumext_store_addto_seq:n { \end{enumext*} }
          }
2376
(\textit{End of definition for } \verb|\_enumext_store_level\_open\_vii: and \verb|\_enumext_store_level\_close\_vii:.)
13.26.7 Function for show marks and position
The function \__enumext_print_keyans_box: NN print a box in the left margin with \l__enumext_mark_-
answer_sym_tl used by the wrap-ans, show-ans and show-pos keys. The function takes two arguments:
```

__enumext_print_keyans_box:NN __enumext_print_keyans_box:cc

__enumext_store_level_open_vii:
__enumext_store_level_close_vii:

75 / 156

```
2387     }
2388     }
2389     \skip_horizontal:n { \dim_use:N #2 }
2390     }
2391 \cs_generate_variant:Nn \__enumext_print_keyans_box:NN { cc }
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_print_keyans_box:NN.)$

13.27 The internal label and ref

The function __enumext_store_internal_ref: handles the "internal label and ref" system used by the save-ref and mark-ref keys for \anskey will allow to execute \ref{\store name: position}} and will return 1.(a).i.A.

__enumext_store_internal_ref:

First we will remove the dots "." from the current $\langle labels \rangle$, we do not want to get double dots in our references, then we will place this in the variable \l_enumext_newlabel_arg_two_tl.

Here we need to analyse the cases where the environment is started with enumext* and if \anskey or anskey* is running alone in it or if it is running in a nested enumext environment within the starting environment.

```
\bool_lazy_all:nT
         {
2405
           { \bool_if_p:N \g__enumext_starred_bool }
2406
           { \int_compare_p:nNn { \l__enumext_level_int } = { 0 } }
2407
2408
         {
           \tl_put_right:Ne \l__enumext_newlabel_arg_two_tl
             { \tl_use:N \l__enumext_label_copy_vii_tl }
2411
         }
       \bool_lazy_all:nT
         {
           { \bool_not_p:n { \g__enumext_standar_bool } }
           { \bool_if_p:N \l__enumext_standar_bool }
           { \int_compare_p:nNn { \l__enumext_level_int } > { 0 } }
2417
         }
2418
         {
           \tl_put_right:Ne \l__enumext_newlabel_arg_two_tl
               \tl_use:N \l__enumext_label_copy_vii_tl
               \int_step_function:nnN { 1 } { \l__enumext_level_int } \__enumext_tmp:n
2425
```

If started with enumext and if \anskey or anskey* is running alone in it or if it is running in a nested enumext* environment within the starting environment.

```
\bool_lazy_all:nT
         {
           { \bool_if_p:N \g__enumext_standar_bool }
           { \int_compare_p:nNn { \l__enumext_level_int } > { 0 } }
           { \int_compare_p:nNn { \l__enumext_level_h_int } = { 0 } }
         }
         {
2432
           \tl_put_right:Ne \l__enumext_newlabel_arg_two_tl
2433
               \tl_use:N \l__enumext_label_copy_i_tl
2435
               \int_step_function:nnN { 2 } { \l__enumext_level_int } \__enumext_tmp:n
2436
2437
2438
       \cs_set:Npn \__enumext_tmp:n ##1
2439
         { \tl_use:c { l__enumext_label_copy_ \int_to_roman:n {##1} _tl } . }
       \bool_lazy_all:nT
```

Now execute the function $\ensuremath{\texttt{_enumext_newlabel:nn}}$ and save the result in the variable $\ensuremath{\texttt{_l_enumext_-write_aux_file_tl}}$ and finally we write in the .aux file.

(End of definition for $_$ enumext_store_internal_ref:.)

13.28 Common functions for \anskey and anskey* environment

__enumext_store_anskey_code:n

2458

The internal function __enumext_store_anskey_code:n first we pass the $\{\langle argument \rangle\}$ to the prop list, then checks the state of the variable \l__enumext_store_ref_key_bool handled by the save-ref key and will call the function __enumext_store_internal_ref: for the "internal label and ref" system. Followed by this if the show-ans or show-pos keys are active we will show the "wrapped" $\{\langle argument \rangle\}$.

```
2467 \cs_new_protected:Npn \__enumext_store_anskey_code:n #1
2468 {
2469  \int_gincr:N \g__enumext_item_anskey_int
2470  \__enumext_store_addto_prop:n {#1}
2471  \bool_if:NT \l__enumext_store_ref_key_bool
2472  {
2473   \__enumext_store_internal_ref:
2474  }
2475  \__enumext_anskey_show_wrap_left:n { #1 }
```

Now we start processing the $\lceil \langle key = val \rangle \rceil$ passed to the command to build our \item in the variable \l_enumext_store_anskey_arg_tl which we will "store" in the sequence. First we clear the variable \l_enumext_store_anskey_arg_tl and process the $\langle keys \rangle$, if the break-col key is present and the command is running under enumext (not in enumext*) we will add \columnbreak and then \item.

```
\tl_clear:N \l__enumext_store_anskey_arg_tl

\text{bool_lazy_and:nnT}

{ \bool_if_p:N \l__enumext_store_columns_break_bool }

{\text{dool_not_p:n { \l_enumext_starred_bool } }

{

\text{tl_put_left:Nn \l_enumext_store_anskey_arg_tl { \columnbreak } }

}

\text{tl_put_right:Nn \l_enumext_store_anskey_arg_tl { \item }}
}
```

If the item-join key is present and the command is running under enumext* we will add $(\langle number \rangle)$ to \l_enumext_store_anskey_arg_tl.

```
\text{\lool_lazy_and:nnT}
{\text{\lool_not_p:n { \l__enumext_starred_bool } }
{\text{\lint_compare_p:nNn { \l__enumext_store_item_join_int } > { 1 } }
}

\text{\lool_not_p:n { \l__enumext_store_item_join_int } > { 1 } }

\text{\lool_tazy_and:nnT}
{
\text{\lool_not_p:n { \l__enumext_store_item_join_int } > { 1 } }
}

\text{\lool_tazy_and:nnT}

\text{\lool_enumext_store_anskey_arg_tl}

\text{\lool_tazy_and:nnT}

\text{\lool_enumext_store_item_join_int }

\text{\lool_exp_not:V \l__enumext_store_item_join_int }

\text{\lool_exp_not:V \l_enumext_store_item_join_int }

\text{\lool_exp_not:V \l_enumext_store_item_join_int }

\text{\lool_exp_not:V \lool_enumext_store_item_join_int }

\text{\lool_enumext_store_item_join_int }

\text{\lool_enumext_store_ite
```

©2024 by Pablo González L

And now we will review the keys item-star, item-sym* and item-pos* and pass them to \l_enumext_-store_anskey_arg_tl along with the $\{\langle argument \rangle\}$ for \anskey or $\langle body \rangle$ for anskey*.

```
\bool_if:NTF \l__enumext_store_item_star_bool
    \tl_put_right:Nn \l__enumext_store_anskey_arg_tl { * }
    \tl_if_empty:NF \l__enumext_store_item_symbol_tl
     {
       \tl_put_right:Ne \l__enumext_store_anskey_arg_tl
         {
            [ \exp_not:V \l__enumext_store_item_symbol_tl ]
     }
    \dim_compare:nT
        \l__enumext_store_item_symbol_sep_dim != \c_zero_dim
      {
        \tl_put_right:Ne \l__enumext_store_anskey_arg_tl
            [ \exp_not:V \l__enumext_store_item_symbol_sep_dim ]
    \tl_put_right:Nn \l__enumext_store_anskey_arg_tl {#1}
 }
  {
    \tl_put_right:Nn \l__enumext_store_anskey_arg_tl {#1}
```

Finally we check if the save-ref key are active along with the hyperref package load, if both conditions are met, it will create the \hyperlink with "symbol" set by mark-ref key and then store in sequence.

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_store_anskey_code:n.)$

__enumext_anskey_show_wrap_arg:n

The function __enumext_anskey_show_wrap_arg:n "wraps" the $\{\langle argument \rangle\}$ passed to \anskey and the $\langle body \rangle$ for anskey* when using the wrap-ans key.

```
^2530 \cs_new_protected:Npn \__enumext_anskey_show_wrap_arg:n #1
2531
       \par
       \bool_if:NTF \l__enumext_starred_bool
2534
              _enumext_print_keyans_box:NN
              \l__enumext_labelwidth_vii_dim \l__enumext_labelsep_vii_dim
         }
         {
              _enumext_print_keyans_box:cc
              { l__enumext_labelwidth_ \__enumext_level: _dim }
              { l__enumext_labelsep_ \__enumext_level: _dim }
2541
2542
       \__enumext_anskey_wrapper:n { #1 }
2543
2544
```

 $(\textit{End of definition for } \verb|_enumext_anskey_show_wrap_arg:n.)$

__enumext_anskey_show_wrap_left:n

The function __enumext_anskey_show_wrap_left:n will show the "mark" defined by the mark-ans key or the "position" of the $\{\langle content \rangle\}$ stored in the prop list when using the show-pos key on the left margin next to the "wraps" $\{\langle argument \rangle\}$ passed to \anskey and the $\langle body \rangle$ in anskey* on the right side when using the show-ans key.

```
2545 \cs_new_protected:Npn \__enumext_anskey_show_wrap_left:n #1
2546 {
©2024 by Pablo González L
```

```
\bool_if:NT \l__enumext_show_answer_bool
              _enumext_anskey_show_wrap_arg:n { #1 }
       \bool_if:NT \l__enumext_show_position_bool
           \tl_set:Ne \l__enumext_mark_answer_sym_tl
             {
                \group_begin:
                \exp_not:N \normalfont
                \exp_not:N \footnotesize [ \int_eval:n
                    \prop_count:c { g__enumext_ \l__enumext_store_name_tl _prop }
                 }
                 1
2561
                \group_end:
2562
2563
           \__enumext_anskey_show_wrap_arg:n { #1 }
2564
2565
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_anskey_show_wrap_left:n.)$

13.29 The command \anskey

Since we will be "storing content" in a list environment within sequences and can (more or less) manage the options passed to each level, it is necessary that we have a little more control over \item when storing.

The \anskey command will cover this point and give it similar behaviour to that of \item in the enumext and enumext* environments executed as follows \anskey[$\langle key = val \rangle$] { $\langle content \rangle$ }.

__enumext_anskey_unknown:n
__enumext_anskey_unknown:nn

First we'll add the keys break-col, item-join, item-star, item-sym* and item-pos*.

```
2567 \keys_define:nn { enumext / anskey }
    {
      break-col .bool_set:N = \l__enumext_store_columns_break_bool,
      break-col .default:n = true,
      break-col .value_forbidden:n = true,
2571
      item-join .int_set:N = \l__enumext_store_item_join_int,
       item-join .value_required:n = true,
       item-star .bool_set:N = \l__enumext_store_item_star_bool,
       item-star .default:n = true,
       item-star .value_forbidden:n = true,
       item-sym* .tl_set:N = \l__enumext_store_item_symbol_tl,
      item-sym* .value_required:n = true,
      item-pos* .dim_set:N = \l__enumext_store_item_symbol_sep_dim,
       item-pos* .value_required:n = true,
                          = { \__enumext_anskey_unknown:n {#1} },
       unknown .code:n
2581
2582
```

The $\langle keys \rangle$ are stored in \l_keys_key_str and the value (if any) is passed as an argument to the function __enumext_anskey_unknown:n.

 $(\textit{End of definition for } \verb|\|_enumext_anskey_unknown:n | and \verb|\|_enumext_anskey_unknown:nn.)$

The \anskey command will only be present when using the save-ans key in enumext and enumext* environments, otherwise it will return an error.

\anske

We will first call the function __enumext_anskey_safe_outer: to be sure where we execute the command, then we will check the state of the variable \l__enumext_check_answers_bool set by the key no-store, if is true we will increment \g__enumext_item_anskey_int for the internal "check answer" system and

©2024 by Pablo González L

execute the function __enumext_anskey_safe_inner:n to ensure that the command is not nested and that the argument is not empty, finally search the $[\langle key = val \rangle]$ and call the function __enumext_store_-anskey_code:n.

```
2597 \NewDocumentCommand \anskey { o +m }
2598
       \__enumext_anskey_safe_outer:
2599
       \group_begin:
         \bool_if:NT \l__enumext_check_answers_bool
             \tl_if_novalue:nF {#1}
               {
                  \keys_set:nn { enumext / anskey } {#1}
               }
             \tl_if_blank:nTF {#2}
               {
                  \msg_error:nn { enumext } { anskey-empty-arg }
                {
                  \__enumext_anskey_safe_inner:
                  \__enumext_store_anskey_code:n {#2}
           }
2615
       \group_end:
2616
     }
2617
```

(End of definition for \anskey. This function is documented on page 13.)

13.29.1 Internal functions for the command

__enumext_anskey_safe_outer:
__enumext_anskey_safe_inner:

The __enumext_store_anskey_safe_outer: function will return the appropriate messages when the command is executed outside the environment in which the save-ans key was activated.

```
2618 \cs_new_protected:Nn \__enumext_anskey_safe_outer:
2619
       \bool_if:NF \l__enumext_store_active_bool
2621
           \msg_error:nnnn { enumext } { anskey-wrong-place }{ anskey }{ enumext }
2622
2623
       \int_compare:nNnT { \l__enumext_keyans_level_int } = { 1 }
           \msg_error:nnnn { enumext } { command-wrong-place }{ anskey }{ keyans }
         }
       \int_compare:nNnT { \l__enumext_keyans_level_h_int } = { 1 }
2628
           \msg_error:nnnn { enumext } { command-wrong-place }{ anskey }{ keyans* }
2620
         }
2631
       \int_compare:nNnT { \l__enumext_keyans_pic_level_int } = { 1 }
2632
         {
2633
           \msg_error:nnnn { enumext } { command-wrong-place }{ anskey }{ keyanspic }
2634
2635
```

The __enumext_anskey_safe_inner: function will first check if the command is nested, if preceded by a not numbered \item or if it is in *math mode* returning the appropriate messages.

```
2637 \cs_new_protected:Nn \__enumext_anskey_safe_inner:
2638
       \int_incr:N \l__enumext_anskey_level_int
2639
       \int_compare:nNnT { \l__enumext_anskey_level_int } > { 1 }
2641
           \msg_error:nn { enumext } { anskey-nested }
         }
2643
       \bool_if:NF \l__enumext_item_number_bool
           \msg_error:nn { enumext } { anskey-unnumber-item }
         }
       \mode_if_math:T
2648
2649
            \msg_error:nne { enumext } { anskey-math-mode } { \c_backslash_str anskey }
2650
2651
     }
2652
```

 $(\textit{End of definition for } \verb|_=enumext_anskey_safe_outer: and \verb|_=enumext_anskey_safe_inner:|)$

©2024 by Pablo González L 80 / 156

13.30 The environment anskey*

Managing *verbatim content* in an environment is quite complicated, I learned that when creating the **scontents** package, so to be able to have support at this point it is best to play a little with the internal code of **scontents** and *hooks*. Some considerations I should have here before implementing this:

- If some package, class or user has defined the environment with the same name somewhere in the document it would be a problem, you would not know what argument has been passed to store-env, if you are using the key print-env or the write-out key, sure, I can detect and modify it within the enumext and enumext* environments, but it would look strange not to have some keys available when running within these environments.
- A better (perhaps a bit paranoid) option is to define it within the environment in which the save-ans key is executed. and have it available only when that key is executed, here I would have absolute control of the \(\lambda keys \rangle \) and I make sure that write-out is not used, then using \(hooks \) after I undefine it and using \(hook \) before I check if it has been created by any package, class or user and I return a error, then the user will have to see how to solve the problem.

__enumext_undefine_anskey_env:

The function __enumext_undefine_anskey_env: will undefine the environment anskey* and will be passed to the function __enumext_execute_after_env: (§13.31) which is executed after the environment in which the key save-ans is active.

```
2653 \cs_new_protected:Nn \__enumext_undefine_anskey_env:
2654 {
2655 \cs_undefine:c { anskey* }
2656 \cs_undefine:c { endanskey* }
2657 \cs_undefine:c { __scontents_anskey*_env_begin: }
2658 \cs_undefine:c { __scontents_anskey*_env_end: }
2659 }
```

Detection of the anskey* environment outside the enumext and enumext* environments.

```
\__enumext_before_env:nn { enumext }
       \bool_lazy_and:nnT
         { \int_compare_p:nNn { \l__enumext_level_int } = { 0 } }
         { \int_compare_p:nNn { \l__enumext_level_h_int } = { 0 } }
         {
           \cs_if_free:cF { __scontents_anskey*_env_begin: }
2666
             {
2667
               \msg_error:nnn { enumext } { anskey-env-error } { anskey* }
2668
         }
     }
   \__enumext_before_env:nn { enumext* }
    {
       \bool_lazy_and:nnT
2674
         { \int_compare_p:nNn { \l__enumext_level_int } = { 0 } }
2675
         { \int_compare_p:nNn { \l__enumext_level_h_int } = { 0 } }
2676
         {
2677
           \cs_if_free:cF { __scontents_anskey*_env_begin: }
2678
             {
2679
               \msg_error:nnn { enumext } { anskey-env-error } { anskey* }
     }
2683
```

Detection of the anskey* environment inside the keyans, keyans* and keyanspic environments, if preceded by a not numbered \item or if it is in *math mode* returning the appropriate messages.

```
2684 \__enumext_before_env:nn { anskey* }
2685
       \int_compare:nNnT { \l__enumext_keyans_level_int } = { 1 }
2686
           \msg_error:nnn { enumext } { anskey-env-wrong }{ keyans }
         }
       \int_compare:nNnT { \l__enumext_keyans_level_h_int } = { 1 }
           \msg_error:nnn { enumext } { anskey-env-wrong } { keyans* }
         }
       \int_compare:nNnT { \l__enumext_keyans_pic_level_int } = { 1 }
2694
2695
           \msg_error:nnn { enumext } { anskey-env-wrong } { keyanspic }
2696
         }
       \bool_if:NF \l__enumext_item_number_bool
```

(End of definition for __enumext_undefine_anskey_env:.)

anskev*

 The function __enumext_anskey_env_make:n creates the environment anskey* (custom version of scontents environment) by setting the initial keys store-env= $\{\langle store\ name \rangle\}$ and print-env=false. To maintain the scope of the environment and that it is only active when the key save-ans is active we will

pass this function to the function __enumext_storing_exec: (§13.25.1) and we will execute it only if the variable \l__enumext_anskey_env_bool is true, with this we prevent it from being executed again when the environment is nested and the key save-ans is active, which returns an error for part of the package

```
scontents.
```

The function __enumext_anskey_env_define_keys: will add the keys break-col, item-join, item-join, item-star, item-sym* and item-pos* and will leave the keys print-env, store-env and write-out undefined. We will apply this function using the *hook* function __enumext_before_env:nn.

```
2716 \cs_new_protected:Nn \__enumext_anskey_env_define_keys:
       \keys_define:nn { scontents / scontents }
2718
         {
           break-col .bool_gset:N = \g__enumext_store_columns_break_bool,
           break-col .default:n = true,
           break-col .value_forbidden:n = true,
           item-join .int_gset:N = \g__enumext_store_item_join_int,
           item-join .value_required:n = true,
2724
           item-star .bool_gset:N = \g__enumext_store_item_star_bool,
           item-star .default:n = true,
2726
           item-star .value_forbidden:n = true,
           item-sym* .tl_gset:N = \g__enumext_store_item_symbol_tl,
           item-sym* .value_required:n = true,
           item-pos* .dim_gset:N = \g__enumext_store_item_symbol_sep_dim,
           item-pos* .value_required:n = true,
           print-env .undefine:,
           store-env .undefine:,
           write-out .undefine:,
           unknown .code:n
                                  = { \__enumext_anskey_env_unknown:n {##1} },
2736
```

The $\langle keys \rangle$ are stored in \l_keys_key_str and the value (if any) is passed as an argument to the function _enumext_anskey_env_unknown:n.

82 / 156

```
2738 \cs_new_protected:Npn \__enumext_anskey_env_unknown:n #1
2739
       \exp_args:NV \__enumext_anskey_env_unknown:nn \l_keys_key_str {#1}
2740
     }
2741
   \cs_new_protected:Npn \__enumext_anskey_env_unknown:nn #1#2
2742
       \tl_if_blank:nTF {#2}
         {
            \msg_error:nnn { enumext } { anskey-env-key-unknown } {#1}
         }
2747
2748
         {
            \msg error:nnnn { enumext } { anskey-env-key-value-unknown } {#1} {#2}
2749
```

The function __enumext_anskey_env_reset_keys: will leave the keys break-col, item-join, item-join, item-star, item-sym* and item-pos* undefined. We will apply this function using the *hook* function __enumext_after_env:nn.

```
2752 \cs_new_protected:Nn \__enumext_anskey_env_reset_keys:
       \keys_define:nn { scontents / scontents }
2754
           break-col .undefine:,
           item-join .undefine:,
           item-star .undefine:,
           item-sym* .undefine:,
           item-pos* .undefine:,
           write-out .code:n
2761
                                    \bool_set_false:N \l__scontents_storing_bool
                                    \bool_set_true:N \l__scontents_writing_bool
2763
                                    \tl_set:Nn \l__scontents_fname_out_tl {##1}
2765
           write-out .value_required:n = true,
           print-env .meta:nn = { scontents } { print-env = ##1 },
           print-env .default:n = true,
           store-env .meta:nn = { scontents } { store-env = ##1 },
           unknown .code:n
                              = { \__scontents_parse_environment_keys:n {##1} },
    }
```

The function __enumext_rescan_anskey_env:n will be responsible for bringing the $\langle body \rangle$ of the environment saved in the sequence \g__scontents_name_ $\langle store\ name \rangle$ _seq to pass it to our sequence and prop list

```
2773 \cs_new_protected:Npn \__enumext_rescan_anskey_env:n #1
2774 {
2775 \group_begin:
2776 \int_set:Nn \tex_newlinechar:D { `\^^J }
2777 \__scontents_rescan_tokens:x
2778 {
2779 \endgroup % This assumes \catcode`\\=0... Things might go off otherwise.
2780 #1
2781 }
2782 }
```

(End of definition for anskey* and others. This function is documented on page 14.)

__enumext_anskey_env_exec:

The function $_$ enumext_anskey_env_exec: will be responsible for processing all the code necessary for the execution of the environment. The first thing will be to add our $\langle keys \rangle$.

```
2783 \cs_new_protected:Nn \__enumext_anskey_env_exec:
2784 {
2785 \__enumext_before_env:nn { anskey* }
2786 {
2787 \__enumext_anskey_env_define_keys:
2788 }
```

Now we will execute our actions after the anskey* environment is closed. We'll fetch the contents of the *environment body* that is now saved in $g_scontents_name_\langle store\ name \rangle_seq$ and store it in the variable $l_enumext_store_anskey_env_tl$ then we execute the rest of the functions.

```
\hook_if_empty:nF {env/anskey*/after}
2789
2790
           \hook_gremove_code:nn {env/anskey*/after} { * }
         }
       \__enumext_after_env:nn { anskey* }
           \__enumext_anskey_env_save_keys:
           \tl_clear:N \l__enumext_store_anskey_env_tl
2796
           \tl_clear:N \l__enumext_store_anskey_opt_tl
           \bool_if:NT \l__enumext_check_answers_bool
2798
               \tl_gset:Ne \l__enumext_store_anskey_env_tl
                 {
                   \seq_item:ce { g__scontents_name_ \l__enumext_store_name_tl _seq } { -1 }
                 }
               \regex_match:nVTF
                 { ^s \ z \ ^s \ u\{c\_scontents\_hidden\_space\_str} \ z \ }
                 \l__enumext_store_anskey_env_tl
```

(End of definition for __enumext_anskey_env_exec:.)

__enumext_anskey_env_save_keys:
__enumext_anskey_env_store:
__enumext_anskey_env_clean_vars:

The function __enumext_anskey_env_save_keys: processing the $[\langle key = val \rangle]$ passed to the environment and save this in the variable \l_enumext_store_anskey_opt_tl. If the break-col key is present and the environment is running under enumext (not in enumext*) we will add the key break-col.

If the item-join key is present and the command is running under enumext* we will add to \l_enumext_store_anskey_opt_tl.

And now we will review the keys item-star, item-sym* and item-pos* and pass them to \l_enumext_-store_anskey_opt_tl.

```
\bool_if:NT \g__enumext_store_item_star_bool
2836
           \tl_put_left:Ne \l__enumext_store_anskey_opt_tl
2837
             {
2838
                ,item-star,
           \tl_if_empty:NF \g__enumext_store_item_symbol_tl
                \tl_put_left:Ne \l__enumext_store_anskey_opt_tl
                  {
                    ,item-sym* = \exp_not:V \g__enumext_store_item_symbol_tl,
2846
             7
2847
           \dim_compare:nT
2848
             {
2849
                \g__enumext_store_item_symbol_sep_dim != \c_zero_dim
             {
                \tl_put_left:Ne \l__enumext_store_anskey_opt_tl
                  {
                    ,item-pos* = \exp_not:V \g__enumext_store_item_symbol_sep_dim,
                  }
             }
2857
          }
2858
```

The function __enumext_anskey_env_store: will be responsible for storing the content of the environment using the functions __enumext_store_anskey_code:n and __enumext_rescan_anskey_env:n.

```
2860 \cs_new_protected:Nn \__enumext_anskey_env_store:
©2024 by Pablo González L
```

```
\group_begin:
         \tl_if_empty:NTF \l__enumext_store_anskey_opt_tl
2862
2864
             \exp_args:Ne
               \__enumext_store_anskey_code:n
                      _enumext_rescan_anskey_env:n { \l__enumext_store_anskey_env_tl }
           }
             \keys_set_known:nV { enumext / anskey } \l__enumext_store_anskey_opt_tl
             \exp_args:Ne
               \__enumext_store_anskey_code:n
                    \__enumext_rescan_anskey_env:n { \l__enumext_store_anskey_env_tl }
2876
2877
2878
       \group_end:
2879
```

The function $\ensuremath{\mbox{\mbox{$\setminus$}}}$ enumext_anskey_env_clean_vars: will return the global variables used by the $\langle \textit{keys} \rangle$ to their initial state.

```
2881 \cs_new_protected:Nn \__enumext_anskey_env_clean_vars:
2882 {
2883    \bool_gset_false:N \g__enumext_store_columns_break_bool
2884    \int_gzero:N    \g__enumext_store_item_join_int
2885    \bool_gset_false:N \g__enumext_store_item_star_bool
2886    \tl_gclear:N    \g_enumext_store_item_symbol_tl
2887    \dim_gzero:N    \g_enumext_store_item_symbol_sep_dim
2888  }
```

 $(\textit{End of definition for } _\texttt{enumext_anskey_env_save_keys:}, \\ \\ _\texttt{enumext_anskey_env_store:}, \\ and \\ \\ \\ _\texttt{enumext_anskey_env_clean_vars:})$

13.31 Executing anskey*, check-ans and write .log

__enumext_execute_after_env:

The __enumext_execute_after_env: function will first return the appropriate message for the end of the environment in which the save-ans key is being executed, then call the __enumext_item_answer_diff: function and then will write the values of the global variables used to the .log file. If the key check-ans is active it will execute the function __enumext_check_ans_show: and show the result in the terminal, otherwise it will execute the function __enumext_check_ans_log: and write the results in the .log file, undefine the environment anskey* (§13.30) through the function __enumext_undefine_anskey_env: and finally we execute the function __enumext_reset_global_vars: returning the used variables to their original state.

```
2889 \cs_new_protected:Nn \__enumext_execute_after_env:
    {
2890
       \int_compare:nNnT { \l__enumext_level_int } = { 0 }
2891
           \tl_if_empty:NF \g__enumext_store_name_tl
2893
               \__enumext_stop_save_ans_msg:
               \__enumext_item_answer_diff:
               \__enumext_log_global_vars:
               \__enumext_log_answer_vars:
               \bool_if:NTF \g__enumext_check_ans_key_bool
                 {
                    \__enumext_check_ans_show:
                  { \__enumext_check_ans_log: }
               \__enumext_undefine_anskey_env:
             _enumext_reset_global_vars:
         }
    }
2008
```

(End of definition for __enumext_execute_after_env:.)

This function is passed to the function __enumext_after_env:nn for the environments enumext (§13.38) and enumext* (§13.43) and it is executed only when the environments are not nested or at some level of these..

©2024 by Pablo González L 85/156

13.32 Common functions for keyans, keyans* and keyanspic

13.32.1 Storing content in prop list

__enumext_keyans_addto_prop:n

The function __enumext_keyans_addto_prop:n will pass the the current $\langle label \rangle$ for \item* in keyans environment and the current $\langle label \rangle$ for \anspic* in keyanspic environment followed by the $\langle contents \rangle$ of the optional argument of both commands to the \l__enumext_store_current_label_tl variable, which will be stored to the prop list defined by the save-ans key using the function __enumext_store_addto_prop: V.

If the *optional argument* is present and the save-sep key is not empty, we save it.

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_keyans_addto_prop:n.)$

13.32.2 The save-ref key for keyans, keyans* and keyanspic

The "internal label and ref" system for the keyans, keyans* and keyanspic environments has slight differences with the one implemented for \anskey basically because in this environments the interest is in the current $\langle label \rangle$ for \item* and \anspic* with the $\langle contents \rangle$ of the optional argument. The mechanism defined here will allow to execute \ref{\langle store name: position}} and will return 1. (A).

__enumext_keyans_store_ref:
 __enumext_keyans_store_ref_aux_i:
 __enumext_keyans_store_ref_aux_ii:

The function __enumext_keyans_store_ref: handles the "internal label and ref" system used by the save-ref key for \item* and \anspic* commands. First we will create copies of the current $\langle labels \rangle$ and remove the dots "." from them, we do not want to get double dots in references.

The auxiliary function __enumext_keyans_store_ref_aux_i: set the variable \l__enumext_newlabel_-arg_one_tl which will contain $\{\langle store\ name: position \rangle\}$ analyzing whether the environment in which they are executed is enumext* or enumext.

```
{
           \tl_put_right:Ne \l__enumext_newlabel_arg_two_tl
             { \l__enumext_label_copy_i_tl . \l__enumext_label_copy_vi_tl }
         }
2957
       \int_compare:nNnT { \l__enumext_keyans_level_int } = { 1 }
2958
2959
           \tl_put_right:Ne \l__enumext_newlabel_arg_two_tl
             { \l__enumext_label_copy_i_tl . \l__enumext_label_copy_v_tl }
       \int_compare:nNnT { \l__enumext_keyans_level_h_int } = { 1 }
           \tl_put_right:Ne \l__enumext_newlabel_arg_two_tl
             { \l__enumext_label_copy_i_tl . \l__enumext_label_copy_viii_tl }
2066
       \tl_put_right:Ne \l__enumext_newlabel_arg_one_tl
2968
         {
2969
           \l__enumext_store_name_tl \c_colon_str
           \int_eval:n { \prop_count:c { g__enumext_ \l__enumext_store_name_tl _prop } }
2971
       \__enumext_keyans_store_ref_aux_ii:
2973
```

Now auxiliary function $\ensuremath{\texttt{_enumext_keyans_store_ref_aux_ii:}}$ save the result in the variable $\ensuremath{\texttt{_l_enumext_write_aux_file_tl}}$ and finally we write in the $\ensuremath{\texttt{_aux}}$ file.

(End of definition for __enumext_keyans_store_ref: , __enumext_keyans_store_ref_aux_i: , and __enumext_keyans_store_ref_aux_ii:.)

13.32.3 Storing content in sequence

__enumext_keyans_addto_seq:n
__enumext_keyans_addto_seq_link:

The function __enumext_keyans_addto_seq:n will pass the contents of the current $\langle label \rangle$ \l__enumext_label_v_tl for the keyans environment and the \l__enumext_label_vi_tl for the keyanspic environment when using \item* and \anspic*, followed by the $\langle contents \rangle$ of the optional argument of both commands to the \l__enumext_store_current_label_tl variable to the sequence defined by the saveans key.

```
<sub>2985</sub> \cs_new_protected:Npn \__enumext_keyans_addto_seq:n #1
       \tl_clear:N \l__enumext_store_current_label_tl
2987
       \int_compare:nNnTF { \l__enumext_keyans_pic_level_int } = { 1 }
2988
2989
           \tl_put_right:Ne \l__enumext_store_current_label_tl {    \item \l__enumext_label_vi_tl }
         }
         {
2992
           \tl_put_right:Ne \l__enumext_store_current_label_tl { \item \l__enumext_label_v_tl }
2993
         }
       \tl_if_novalue:nF { #1 }
         {
           \tl_if_empty:NF \l__enumext_store_keyans_item_opt_sep_tl
2998
             {
               \tl_put_right:Ne \l__enumext_store_current_label_tl
                 {
                    \l__enumext_store_keyans_item_opt_sep_tl
           \tl_put_right:Ne \l__enumext_store_current_label_tl { #1 }
       \__enumext_keyans_addto_seq_link:
     }
```

Checks if the save-ref key is active along with the hyperlink and then store using the __enumext_store_addto_seq:V function. Finally, copy the contents of the variable \l__enumext_store_current_label_tl into the global variable \g__enumext_check_ans_item_tl to be used by the function __enumext_check_starred_cmd:n and

increment the value of the integer variable $\g_{\text{enumext_item_anskey_int}}$ handled by the check-anskev.

```
3008 \cs_new_protected:Nn \__enumext_keyans_addto_seq_link:
    {
       \bool_lazy_and:nnT
         { \bool_if_p:N \l__enumext_store_ref_key_bool }
         { \bool_if_p:N \l__enumext_hyperref_bool }
3013
           \tl_put_right:Ne \l__enumext_store_current_label_tl
3014
             {
3015
               \hfill \exp_not:N \hyperlink
3016
                 {
3017
                    \exp_not:V \l__enumext_newlabel_arg_one_tl
                 { \exp_not:V \l__enumext_mark_ref_sym_tl }
             }
         }
         _enumext_store_addto_seq:V \l__enumext_store_current_label_tl
       \bool_if:NT \l__enumext_check_answers_bool
           \int_gincr:N \g__enumext_item_anskey_int
3026
3027
3028
```

 $(\textit{End of definition for } \verb|_=enumext_keyans_addto_seq:n | and \verb|_=enumext_keyans_addto_seq_link:|)$

13.32.4 The show-ans and show-pos keys for keyans and keyanspic

The code is very similar to the \anskey code, but, if I change the order of the operations the counter off $\langle label \rangle$ are incorrect.

__enumext_keyans_show_left:n
__enumext_keyans_show_ans:
__enumext_keyans_show_pos:
__enumext_keyans_show_item_opt:

Common function to show *starred commands* \item* and $\langle position \rangle$ of stored content in *prop list* for keyans and keyanspic. Need add 1 to \g__enumext_ $\langle store\ name \rangle$ _prop for show-pos key.

```
3029 \cs_new_protected:Npn \__enumext_keyans_show_left:n #1
3030
       \tl_if_novalue:nF { #1 }
         {
            \tl_set:Ne \l__enumext_store_current_opt_arg_tl { #1 }
       \bool_if:NT \l__enumext_show_answer_bool
3035
         {
3036
            \__enumext_keyans_show_ans:
3037
3038
       \bool_if:NT \l__enumext_show_position_bool
3039
            \__enumext_keyans_show_pos:
3041
3043
   \cs_new_protected:Nn \__enumext_keyans_show_item_opt:
       \tl_if_empty:NF \l__enumext_store_current_opt_arg_tl
3046
3047
            \bool_lazy_or:nnT
3048
              { \bool_if_p:N \l__enumext_show_answer_bool }
              { \bool_if_p:N \l__enumext_show_position_bool }
3050
3051
                \__enumext_keyans_wrapper_opt:n { \l__enumext_store_current_opt_arg_tl } \c_space_tl
3053
         }
     }
3055
3056 \cs_new_protected:Nn \__enumext_keyans_show_ans:
3057
       \bool_if:NT \l__enumext_starred_bool
3058
3059
            \dim_set_eq:NN \l__enumext_labelwidth_i_dim \l__enumext_labelwidth_vii_dim
            \dim_set_eq:NN \l__enumext_labelsep_i_dim \l__enumext_labelsep_vii_dim
       \tl_put_left:Nn \l__enumext_label_v_tl
              _enumext_print_keyans_box:NN
              \l__enumext_labelwidth_i_dim \l__enumext_labelsep_i_dim
©2024 by Pablo González L
                                                                                                    88 / 156
```

```
\cs_new_protected:Nn \__enumext_keyans_show_pos:
       \bool_if:NT \l__enumext_starred_bool
3071
3072
           \dim_set_eq:NN \l__enumext_labelwidth_i_dim \l__enumext_labelwidth_vii_dim
3073
           \dim_set_eq:NN \l__enumext_labelsep_i_dim \l__enumext_labelsep_vii_dim
3074
3075
       \int_compare:nNnTF { \l__enumext_keyans_pic_level_int } = { 1 }
3076
           \tl_set:Ne \l__enumext_mark_answer_sym_tl
             {
               \group_begin:
               \exp_not:N \normalfont
               \exp_not:N \footnotesize [ \int_eval:n
3082
                 {
3083
                    \prop_count:c { g__enumext_ \l__enumext_store_name_tl _prop }
               \group_end:
         }
         {
           \tl_set:Ne \l__enumext_mark_answer_sym_tl
               \group_begin:
               \exp_not:N \normalfont
               \exp_not:N \footnotesize [ \int_eval:n
                   \prop_count:c { g__enumext_ \l__enumext_store_name_tl _prop } + 1
                 }
                 1
               \group_end:
         }
       \tl_put_left:Nn \l__enumext_label_v_tl
3104
             _enumext_print_keyans_box:NN
             \l__enumext_labelwidth_i_dim \l__enumext_labelsep_i_dim
         }
```

 $(\textit{End of definition for } \verb|_-enumext_keyans_show_left:n and others.)$

13.33 Redefining \item and \makelabel in enumext

Redefining the \item command is not as simple as I thought. This command works in conjunction with the \makelabel command so I have to redefine both of them, in addition to this, we will have to use a couple of global variables to pass the values from one command to the other.

When *labeling* PDF is active \makelabel is redefined as \hss #1 and the only way to get the align key to work correctly is to redefine \makelabel using \makebox. The best way to implement this is to use the conditional command \IfDocumentMetadataTF to force this redefinition and the dedicated mode-box key to manually activate it by the user.

The \item and \item[$\langle custom \rangle$] commands work in the usual way on enumext and we will add \item*, \item*[$\langle symbol \rangle$] and \item*[$\langle symbol \rangle$][$\langle offset \rangle$].

__enumext_default_item:n

First we will see if the *optional argument* is present, if it is NOT present we will check the state of the variable \l__enumext_check_answers_bool set by the key no-store, set the boolean variable \l__enumext_-wrap_label_X_bool to "true" for the key wrap-label and execute __enumext_item_std:w and the key itemindent, otherwise we will check the state of the boolean variable \l__enumext_wrap_label_opt_-X_bool set by the key wrap-label* and execute __enumext_item_std:w with the *optional argument* and the key itemindent.

```
3100 \cs_new_protected:Npn \__enumext_default_item:n #1
3110 {
3111 \tl_if_novalue:nTF {#1}
3112 {
3113 \bool_if:NT \l__enumext_check_answers_bool
3114 {
3115 \int_gincr:N \g__enumext_item_number_int
3116 \bool_set_true:N \l__enumext_item_number_bool
```

©2024 by Pablo González L

```
}

bool_set_true:c { l__enumext_wrap_label_ \__enumext_level: _bool }

\__enumext_item_std:w \tl_use:c { l__enumext_fake_item_indent_ \__enumext_level: _tl }

}

bool_set_eq:cc
{ l__enumext_wrap_label_ \__enumext_level: _bool }

{ l__enumext_wrap_label_opt_ \__enumext_level: _bool }

\__enumext_item_std:w [#1] \tl_use:c { l__enumext_fake_item_indent_ \__enumext_level: _tl
}

}

}

}

}

**The content of the content o
```

__enumext_starred_item:nn
__enumext_item_star_exec:

The $\idesign* (symbol) and <math>\idesign* (symbol) [(offset)] works like the numbered <math>\idesign* (symbol) to the "left" of the (label) separated from it by the value the second optional argument (offset).$

```
#1: \l_enumext_item_symbol_X_tl
#2: \l_enumext_item_symbol_sep_X_dim
```

(End of definition for __enumext_default_item:n.)

First we will make a copy of \l__enumext_item_symbol_X_tl which is set by the key item-sym* or passed as "first" optional argument in the global variable \g__enumext_item_symbol_aux_tl, followed by setting the variable \l__enumext_item_symbol_sep_X_dim set by the key item-pos* or by the "second" optional argument, then we will see the state of the variable \l__enumext_check_answers_bool set by the key no-store, set the boolean variable \l__enumext_wrap_label_X_bool to "true" for the key wrap-label and execute __enumext_item_std:w and the key itemindent.

```
3128 \cs_new_protected:Npn \__enumext_starred_item:nn #1 #2
3129
       \tl_if_novalue:nTF {#1}
3130
         {
           \tl_gset_eq:Nc
             \g__enumext_item_symbol_aux_tl { l__enumext_item_symbol_ \__enumext_level: _tl }
           \tl_gset:Nn \g__enumext_item_symbol_aux_tl {#1}
         }
       \tl_if_novalue:nTF {#2}
         {
           \dim_set_eq:cc
             { l__enumext_item_symbol_sep_ \__enumext_level: _dim }
             { l__enumext_labelsep_ \__enumext_level: _dim }
         }
         {
           \dim_set:cn { l__enumext_item_symbol_sep_ \__enumext_level: _dim } {#2}
         }
       \bool_if:NT \l__enumext_check_answers_bool
         {
           \int_gincr:N \g__enumext_item_number_int
3149
           \bool_set_true:N \l__enumext_item_number_bool
       \bool_set_true:c { l__enumext_wrap_label_ \__enumext_level: _bool }
       \__enumext_item_std:w \tl_use:c { l__enumext_fake_item_indent_ \__enumext_level: _tl }
3154
```

The function __enumext_item_star_exec: will be responsible for executing \item* for the enumext environment.

 $(\textit{End of definition for } \verb|_=enumext_starred_item:nn and \verb|_=enumext_item_star_exec:|)$

__enumext_redefine_item:

The function __enumext_redefine_item: will redefine the \item command in the enumext environment adding \item*. This function are passed to __enumext_list_arg_two_X: used in the definition of the enumext environment (§13.38).

__enumext_make_label_std:

__enumext_make_label_box:

```
\cs_new_protected:Nn \__enumext_redefine_item:
                         3166
                                 \RenewDocumentCommand \item { s o o }
                         3167
                         3168
                                     \bool_if:nTF {##1}
                         3169
                                       {
                                         \__enumext_starred_item:nn {##2} {##3}
                                         \__enumext_default_item:n {##2} }
                                   }
                              }
                         (End of definition for \__enumext_redefine_item:.)
                         The function \__enumext_make_label: redefine \makelabel for the keys mode-box, align, font, wrap-
\__enumext_make_label:
                         label, wrap-label* and \item* for enumext environment. This function are passed to \__enumext_-
                         list_arg_two_X: used in the definition of the enumext environment (§13.38).
                         3176 \cs_new_protected:Nn \__enumext_make_label:
                         3178
                                 \IfDocumentMetadataTF
                         3179
                                     \__enumext_make_label_box:
                                   }
                                   {
                                     \bool_if:NTF \l__enumext_mode_box_bool
                                             _enumext_make_label_box:
                         3185
                         3186
                                       {
                                          \__enumext_make_label_std:
                                   }
                         Standard definition when \DocumentMetadata is not active.
                         3192 \cs_new_protected:Nn \__enumext_make_label_std:
                                \RenewDocumentCommand \makelabel { m }
                                     \tl_use:c { l__enumext_label_fill_left_ \__enumext_level: _tl }
                         3196
                                     \tl_use:c { l__enumext_label_font_style_ \__enumext_level: _tl }
                                     \bool_if:cTF { l__enumext_wrap_label_ \__enumext_level: _bool }
                         3198
                         3199
                                         \__enumext_item_star_exec:
                                         \use:c { __enumext_wrapper_label_ \__enumext_level: :n } { ##1 }
                                       }
                                       { ##1 }
                                     \tl_use:c { l__enumext_label_fill_right_ \__enumext_level: _tl }
                                     \tl_gclear:N \g__enumext_item_symbol_aux_tl
                              }
                         3207
                         Definition using \makebox when \DocumentMetadata is active or mode-box is active.
                         3208 \cs_new_protected:Nn \__enumext_make_label_box:
                         3209
                                 \RenewDocumentCommand \makelabel { m }
                                     \makebox
                                       [ \dim_use:c { l__enumext_labelwidth_ \__enumext_level: _dim } ]
                                       [\str_use:c { l__enumext_align_label_pos_ \__enumext_level: _str } ]
                                         \tl_use:c { l__enumext_label_font_style_ \__enumext_level: _tl }
                                         \bool_if:cTF { l__enumext_wrap_label_ \__enumext_level: _bool }
                         3218
                                             \ enumext item star exec:
                                             \use:c { __enumext_wrapper_label_ \__enumext_level: :n } { ##1 }
                                           }
                                         \tl_gclear:N \g__enumext_item_symbol_aux_tl
                                   }
```

 $(\textit{End of definition for } \c make_label:, \c enumext_make_label:, \c enumext_make_label_std:, and \c enumext_make_label_box:.)$

13.34 Setting item-sym* and item-pos* keys

In order to have a cleaner implementation of $\forall item^*$ for the enumext and enumext* environments it is best to define a couple of keys that allow us to control and set by default the $\langle symbol \rangle$ and its $\langle offset \rangle$.

Define and set item-sym* and item-pos* keys for enumext and enumext*. item-svm* item-pos* 3227 \cs_set_protected:Npn __enumext_tmp:nn #1 #2 3228 \keys_define:nn { enumext / #1 } 3229 { 3230 item-sym* .tl_set:c = { l__enumext_item_symbol_#2_tl }, 3231 item-sym* .value_required:n = true, 3232 item-sym* .initial:n = {\textasteriskcentered}, 3233 item-pos* .dim_set:c = { l__enumext_item_symbol_sep_#2_dim }, item-pos* .value_required:n = true, ₃₂₃₈ \clist_map_inline:nn {level-1}{i}, {level-2}{ii}, {level-3}{iii}, {level-4}{iv}, {enumext*}{vii} 3240 } 3241 { __enumext_tmp:nn #1 } 3242

(End of definition for item-sym* and item-pos*.)

13.35 Handling unknown keys

At this point in the code I already know that I will not add more $\langle keys \rangle$ and since I have already been quite *paranoid and restrictive* with the definitions of environments and commands, the only thing left to do is do it with the $\langle keys \rangle$ (you have to be consistent in life).

13.35.1 Handling unknown keys for keyans, keyans* and keyanspic

Define and set unknown key for keyans, keyans* and keyanspic environments.

__enumext_keyans_unknown_keys:n __enumext_keyans_unknown_keys:nn

unknown

Internal functions for handling unknown key.

```
3251 \cs_new_protected:Npn \__enumext_keyans_unknown_keys:n #1
3252
       \exp_args:NV \__enumext_keyans_unknown_keys:nn \l_keys_key_str {#1}
3255 \cs_new_protected:Npn \__enumext_keyans_unknown_keys:nn #1#2
3256
       \tl_if_blank:nTF {#2}
3257
           \msg_error:nnn { enumext } { keyans-unknown-key } {#1}
         }
         {
3261
           \msg_error:nnnn { enumext } { keyans-unknown-key-value } {#1} {#2}
3262
         }
3263
3264
```

 $(\textit{End of definition for unknown}\ , \ __enumext_keyans_unknown_keys:n , and \ __enumext_keyans_unknown_keys:n.)$

13.35.2 Handling unknown keys for enumext*

unknown
__enumext_starred_unknown_keys:n
__enumext_starred_unknown_keys:nn

Define and set unknown key for enumext* environment.

©2024 by Pablo González L 92/156

Internal functions for handling unknown key.

```
3269 \cs_new_protected:Npn \__enumext_starred_unknown_keys:n #1
       \exp_args:NV \__enumext_starred_unknown_keys:nn \l_keys_key_str {#1}
3271
     }
3272
3273 \cs_new_protected:Npn \__enumext_starred_unknown_keys:nn #1#2
3274
       \tl_if_blank:nTF {#2}
3276
           \msg_error:nnn { enumext } { starred-unknown-key } {#1}
3277
         }
3278
         {
            \msg_error:nnnn { enumext } { starred-unknown-key-value } {#1} {#2}
     }
3282
```

 $(\textit{End of definition for unknown}, \verb|\|_enumext_starred_unknown_keys:n, and \verb|\|_enumext_starred_unknown_keys:nn.|)$

13.35.3 Handling unknown keys for enumext

unknown

__enumext_standar_unknown_keys:n

__enumext_standar_unknown_keys:nn

Defines and set the key unknown for enumext environment.

Internal functions for handling unknown key.

```
3291 \cs_new_protected:Npn \__enumext_standar_unknown_keys:n #1
3292
       \exp_args:NV \__enumext_standar_unknown_keys:nn \l_keys_key_str {#1}
3293
     }
3294
3295 \cs_new_protected:Npn \__enumext_standar_unknown_keys:nn #1#2
       \tl_if_blank:nTF {#2}
3298
         {
            \msg_error:nnn { enumext } { standar-unknown-key } {#1}
3299
         }
3300
         {
3301
            \msg_error:nnnn { enumext } { standar-unknown-key-value } {#1} {#2}
3303
```

 $(End\ of\ definition\ for\ unknown\ , \ _enumext_standar_unknown_keys:n\ , and\ \setminus _enumext_standar_unknown_keys:nn.)$

13.36 Redefining \item and \makelabel in keyans

The \item and \item[$\langle custom \rangle$] commands work in the usual way in keyans, but the \item* and \item*[$\langle content \rangle$] commands store the current $\langle label \rangle$ next to the $\langle content \rangle$ if it is present in the sequence and prop list defined by save-ans key.

__enumext_keyans_default_item:n

The function __enumext_keyans_default_item:n executes the original behavior of the \item along with the keys wrap-label, wrap-label* and itemindent.

(End of definition for __enumext_keyans_default_item:n.)

©2024 by Pablo González L 93/156

__enumext_keyans_starred_item:n

The function __enumext_keyans_starred_item:n which will make a temporary copy of the current $\langle label \rangle$, execute the show-ans or show-pos keys using the function __enumext_keyans_show_left:n and will display the $\langle contents \rangle$ of that item using the internal copy __enumext_item_std:w, this is necessary to prevent incrementing the current "counter" of the original $\langle label \rangle$, followed by this it will execute function __enumext_keyans_show_item_opt: handled by wrap-opt key.

Recover the original value of the current $\langle label \rangle$ and *store* it first in the *prop list* (including the *optional argument*), run the internal "label and ref" system if the save-ref key is active, *store* it in the *sequence* and finally increments \g_enumext_check_starred_cmd_int for internal check system.

(End of definition for $_$ enumext_keyans_starred_item:n.)

\item*
__enumext_keyans_redefine_item:

The function __enumext_keyans_redefine_item: is responsible for adding the *starred argument* and *optional argument* by the __enumext_list_arg_two_v: function in the definition of the keyans environment. Here we need to use \peek_remove_spaces:n to prevent an unwanted space when using \item* in conjunction with the itemindent key. This function are passed to __enumext_list_arg_two_v: used in the definition of the keyans environment (§13.37.2).

(End of definition for \item* and __enumext_keyans_redefine_item:. This function is documented on page 15.)

__enumext_keyans_make_label:
__enumext_keyans_make_label_std:
__enumext_keyans_make_label_box:

The function __enumext_keyans_make_label: redefine \makelabel for the keys mode-box, align, font, wrap-label, wrap-label* and \item* for keyans environment. This function are passed to __enumext_list_arg_two_v: used in the definition of the keyans environment (§13.37.2).

©2024 by Pablo González L 94/156

Standard definition when \DocumentMetadata is not active.

Definition using \makebox when \DocumentMetadata is active or mode-box is active.

 $(End\ of\ definition\ for\ __enumext_keyans_make_label:,\ __enumext_keyans_make_label_std:,\ and\ __enumext_keyans_make_label_box:.)$

13.37 Second argument of the lists

At this point of the code we have already programmed most the necessary tools to create a custom list environment, remember that the function __enumext_start_list:nn takes two arguments, the first one we have ready, the second one we will define for all the levels of the environment enumext and the environment keyans.

13.37.1 Calculation of \leftmargin and \itemindent

Consider the figure 9 where the default margins (on the left) of a list are represented.



Figure 9: Representation of standard horizontal lengths in list environment.

The idea is to have control over these margins so that our list does not overlap the left margin of the page. The *key* relationship is that the right edge of the \labelsep equals the right edge of the \itemindent, so that the left edge of the *label box* is at \leftmargin+\itemindent minus \labelwidth+\labelsep. Thus, the handling of the margins by the package will be as shown in the figure 10.



Figure 10: Representation of horizontal lengths concept in list in enumext.

Where the default values will look like in the figure 11.

__enumext_calc_hspace:NNNNNNN\ _enumext_calc_hspace:cccccc The function __enumext_calc_hspace: NNNNNNN takes seven arguments to be able to determine horizontal spaces for all list environment:

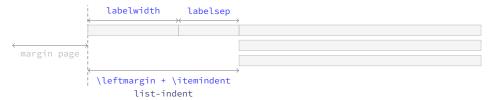


Figure 11: Default horizontal lengths in enumext.

```
#1: \l__enumext_labelwidth_X_dim #2: \l__enumext_labelsep_X_dim
#3: \l__enumext_listoffset_X_dim #4: \l__enumext_leftmargin_tmp_X_dim
#5: \l__enumext_leftmargin_X_dim #6: \l__enumext_itemindent_X_dim
#7: \l__enumext_leftmargin_tmp_X_bool
```

And returns the "adjusted" values of \leftmargin and \itemindent.

This function is passed to __enumext_list_arg_two_X: which is used in the definition of the enumext and keyans environments (§13.37.2).

If no value has been passed to the labelwidth and labelsep keys we set the default values for $\l_--enumext_leftmargin_tmp_X_dim$.

```
\bool_if:nF #7 { \dim_set:Nn #4 { #1 + #2} }
```

We now analyze the cases and set the values for \leftmargin and \itemindent.

```
\dim_compare:nNnTF { #4 } < { \c_zero_dim }</pre>
         {
            \dim_set:Nn #6 { #1 + #2 - #4}
            \dim_set:Nn #5 { #1 + #2 + #3 - #6 }
         }
3408
         {
            \dim_{compare:nNnT { #4 } = { #1 + #2 }
3410
              { \dim_set:Nn #6 { \c_zero_dim } }
            \dim_compare:nNnT { #4 } < { #1 + #2 }
              { \dim_set:Nn #6 { #1 + #2 - #4} }
            \dim_{compare:nNnT { #4 } > { #1 + #2 }
              {
3415
                \dim_set:Nn #6 { -#1 - #2 + #4}
3416
                \dim_set:Nn #6 { #6*-1}
3417
            \dim_set:Nn #5 { #1 + #2 + #3 - #6 }
3419
3420
3421
3422 \cs_generate_variant:Nn \__enumext_calc_hspace:NNNNNNN { cccccc }
```

(End of definition for $\ensuremath{\setminus} = enumext_calc_hspace: NNNNNNN.$)

13.37.2 Setting second argument of the lists

__enumext_list_arg_two_i: We will "not set" \leftmargini, \leftmarginii, \leftmarginiii or \leftmarginiv, in this case, we __enumext_list_arg_two_ii: will directly set the parameters for vertical and horizontal list spacing per level.

©2024 by Pablo González L 96/156

```
\clist_map_inline:nn
              { labelsep, labelwidth, itemindent, leftmargin, rightmargin, listparindent }
              { \dim_set_eq:cc {####1} { l__enumext_###1_#1_dim } }
           \clist_map_inline:nn { topsep, parsep, partopsep, itemsep }
              { \skip_set_eq:cc {####1} { l__enumext_####1_#1_skip } }
           \usecounter { enumX#1 }
           \setcounter { enumX#1 } { \int_eval:n { \int_use:c { l__enumext_start_#1_int } - 1 } }
3438
           \str_if_eq:nnTF {#1} { v }
                \__enumext_keyans_redefine_item:
                \__enumext_keyans_make_label:
                \__enumext_keyans_ref:
                \__enumext_keyans_fake_item_indent:
                \bool_if:cT { l__enumext_show_length_#1_bool }
                  {
                    \msg_term:nnnn { enumext } { list-lengths-not-nested } { v } { keyans }
3447
3448
             }
3449
3450
                \__enumext_redefine_item:
                \__enumext_make_label:
                \__enumext_standar_ref:
                \__enumext_fake_item_indent:
                \bool_if:cT { l__enumext_show_length_#1_bool }
                  {
                    \msg_term:nnne { enumext } { list-lengths } {#1}
                      { \int_use:N \l__enumext_level_int }
3458
                  }
3459
             }
         }
3463 \clist_map_inline:nn { i, ii, iii, iv, v } { \__enumext_tmp:n {#1} }
(End of definition for \__enumext_list_arg_two_i: and others.)
```

__enumext_list_arg_two_vii:
 __enumext_list_arg_two_viii:

For the horizontal environments enumext* and keyans* the implementation is similar, but, the value of \partopsep is always opt. At this point we will modify the parsep key to make it take the value of the itemsep key and later, in the environment definition, we will modify parindent to make it set the value of lisparindent and parsep to set the value of \parskip locally.

```
lisparindent and parsep to set the value of \parskip locally.
3464 \cs_set_protected:Npn \__enumext_tmp:n #1
3465
     {
       \cs_new_protected:cpn { __enumext_list_arg_two_#1: }
3466
3467
           \bool_set_true:c { l__enumext_leftmargin_tmp_#1_bool }
3468
           \dim_zero:c { l__enumext_leftmargin_tmp_#1_dim }
           \__enumext_calc_hspace:cccccc
3470
             { l__enumext_labelwidth_#1_dim } { l__enumext_labelsep_#1_dim }
3471
             { l__enumext_listoffset_#1_dim } { l__enumext_leftmargin_tmp_#1_dim }
3472
             { l__enumext_leftmargin_#1_dim } { l__enumext_itemindent_#1_dim }
             { l__enumext_leftmargin_tmp_#1_bool }
           \clist_map_inline:nn
             { labelsep, labelwidth, itemindent, leftmargin, rightmargin, listparindent }
             { \dim_set_eq:cc {####1} { l__enumext_####1_#1_dim } }
3477
           \clist_map_inline:nn { topsep, parsep, partopsep, itemsep }
3478
             { \skip_set_eq:cc {####1} { l__enumext_####1_#1_skip } }
           \skip_set_eq:Nc \parsep { l__enumext_itemsep_#1_skip }
           \skip_zero:N \partopsep
           \usecounter { enumX#1 }
           \setcounter { enumX#1 } { \int_eval:n { \int_use:c { l__enumext_start_#1_int } - 1 } }
           \__enumext_starred_ref:
           \str_if_eq:nnTF {#1} { vii }
                 _enumext_fake_item_indent_vii:
               \bool_if:cT { l__enumext_show_length_vii_bool }
3488
                  { \msg_term:nnnn { enumext } { list-lengths-not-nested } { vii } { enumext* } }
               \__enumext_fake_item_indent_viii:
               \bool_if:cT { l__enumext_show_length_#1_bool }
                  { \msg_term:nnnn { enumext } { list-lengths-not-nested } { #1 } { keyans* } }
©2024 by Pablo González L
                                                                                                 97 / 156
```

```
3496    }
3497    }
3498 \clist_map_inline:nn { vii, viii } { \__enumext_tmp:n {#1} }

(End of definition for \__enumext_list_arq_two_vii: and \__enumext_list_arq_two_viii:)
```

13.38 The environment enumext

__enumext_safe_exec:

The __enumext_safe_exec: function first call the function __enumext_internal_mini_page: to create the environment __enumext_mini_page, then the function __enumext_is_not_nested: which sets \g__enumext_standar_bool to "true" if we are not nested within enumext*, we will increment \l__enumext_level_int to restrict nesting of the environment, set \l__enumext_standar_bool to "true" and finally call the function __enumext_is_on_first_level: which sets \l__enumext_standar_first_bool to "true" only if the environment is not nested and we are at the "first level".

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_safe_exec:.)$

__enumext_parse_keys:n

The __enumext_parse_store_keys:n function first we will clear the variable \l__enumext_series_str used by the key series and then we check if we are at the "first level", if so we process the $\langle keys \rangle$ and then execute the function __enumext_parse_series:n used by the key series and call the function __enumext_nested_base_line_fix: used by the key base-fix, otherwise we will pass the $\langle keys \rangle$ to the inner levels of the environment then we execute the function __enumext_store_active_keys:n and reprocess the $\langle keys \rangle$ to pass them to the sequence if the key save-key is not active.

(End of definition for $_$ enumext_parse_keys:n.)

__enumext_start_store_level:

The __enumext_start_store_level: function activate the "storing structure" mechanism in the sequence for the command \anskey and the environment anskey*.

If enumext are nested in enumext* add __enumext_store_level_open: to preserve the "storing structure".

(End of definition for __enumext_start_store_level:.)

__enumext_stop_store_level:

The __enumext_stop_store_level: function stop the "storing structure" mechanism in the sequence for the command \anskey and the environment anskey*.

(End of definition for __enumext_stop_store_level:.)

__enumext_multicols_start:

The function __enumext_multicols_start: will start the multicols environment according to the value passed by the columns key, then set the default value for \columnsep when columns-sep=0pt and set the value of \multicolsep equal to zero and leave \columnseprule equal to zero for inner levels.

```
3564 \cs_new_protected:Nn \__enumext_multicols_start:
3566
       \int_compare:nNnT
         { \int_use:c { l__enumext_columns_ \__enumext_level: _int } } > { 1 }
3567
3568
           \dim_compare:nNnT
             { \dim_use:c { l__enumext_columns_sep_ \__enumext_level: _dim } } = { \c_zero_dim }
             {
               \dim_set:cn { l__enumext_columns_sep_ \__enumext_level: _dim }
                   ( \dim_use:c { l__enumext_labelwidth_ \__enumext_level: _dim }
                     + \dim_use:c { l__enumext_labelsep_ \__enumext_level: _dim }
                   ) / \int_use:c { l__enumext_columns_ \__enumext_level: _int }
                    - \dim_use:c { l__enumext_listoffset_ \__enumext_level: _dim }
                 }
           \dim_set_eq:Nc \columnsep { l__enumext_columns_sep_ \__enumext_level: _dim }
3580
           \int_compare:nNnT { \l__enumext_level_int } > { 1 }
3581
             {
3582
               \dim_zero:N \columnseprule
3583
```

We will calculate the *vertical spacing* settings for the multicols environment using the function __enumext_-multi_addvspace:, apply our "*vertical adjust spacing*", then start the multicols environment.

(End of definition for __enumext_multicols_start:.)

©2024 by Pablo González L 99/156

__enumext_multicols_stop:

The function __enumext_multicols_stop: will stop the multicols environment and apply our "vertical adjust" spacing. For compatibility with tagged PDF, the closing of the list environment is executed here along with __enumext_stop_store_level:.

```
3594 \cs_new_protected:Nn \__enumext_multicols_stop:
3595
3596
       \int_compare:nNnTF
         { \int_use:c { l__enumext_columns_ \__enumext_level: _int } } > { 1 }
3597
3598
           \__enumext_stop_list:
           \__enumext_stop_store_level:
           \end{multicols}
           \__enumext_unskip_unkern:
           \ enumext unskip unkern:
           \par\addvspace{ \skip_use:c { l__enumext_multicols_below_ \__enumext_level: _skip } }
3604
            \__enumext_stop_list:
           \__enumext_stop_store_level:
     }
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_multicols_stop:|)$

__enumext_before_list:

The function __enumext_before_list: first calls the function __enumext_vspace_above: used by the keys above and above*, then calls the function __enumext_before_args_exec: used by the key before* and finally execute the function __enumext_check_ans_active: for the check answer mechanism.

When the mini-env key is active it will set the value of the \l__enumext_minipage_right_X_dim to be the width of the __enumext_minipage environment on the "right side", using this value together with the value of the \l__enumext_minipage_hsep_X_dim set by the mini-sep key, the value of \l__enumext_minipage_left_X_dim will be set, which will be the width of __enumext_mini_page environment on the "left side", always having a current \linewidth as maximum width between them.

The boolean variable \l__enumext_minipage_active_X_bool will be activated and the integer variable \g__enumext_minipage_stat_int used by the \miniright command will be incremented, then the function __enumext_minipage_add_space: is called and the __enumext_mini_page environment on the "left side" will be initialized followed by the "vertical spacing" applied to preserve the "baseline" between the left and right side environments. After these actions, the function __enumext_multicols_start: is called to handle the multicols environment.

(End of definition for __enumext_before_list:.)

__enumext_second_part:

The function __enumext_second_part: first check the state of the boolean variable \l__enumext_minipage_active_X_bool, if it is "true" a small test will be executed to check if we have omitted the use of \miniright (the __enumext_mini_page environment has not been closed), then close __enumext_mini_page and add the adjusted vertical space \l__enumext_minipage_after_skip, otherwise we will close the multicols environment.

```
3633 \cs_new_protected:Nn \__enumext_second_part:
©2024 by Pablo González L
```

Now we will execute the functions __enumext_after_stop_list: used by the key after, __enumext_-check_ans_key_hook: used by the key check-ans, __enumext_vspace_below: used by the keys below and below*. Finally set \l__enumext_standar_bool to false and call the function __enumext_resume_-save_counter: used by the series, resume and resume* keys.

```
\__enumext_after_stop_list:
3650 \__enumext_check_ans_key_hook:
\__enumext_vspace_below:
\bool_set_false:N \l__enumext_standar_bool
\_enumext_resume_save_counter:
3654 }
```

(End of definition for __enumext_second_part:.)

__enumext_set_item_width:

The function __enumext_set_item_width: will set the value of \itemwidth taking into account the value established by the list-offset key for each level of the environment.

```
3655 \cs_new_protected:Nn \__enumext_set_item_width:
3656
       \dim_set:Nn \itemwidth { \linewidth }
3657
       \dim_compare:nT
         {
           \dim_use:c { l__enumext_listoffset_ \__enumext_level: _dim } != \c_zero_dim
3660
         }
3661
         {
3662
           \dim_sub:Nn \itemwidth
3663
3664
                \dim_use:c { l__enumext_listoffset_ \__enumext_level: _dim }
         }
     }
```

(End of definition for __enumext_set_item_width:.)

enumext Now create the enumext environment based on list environment by levels.

```
3669 \NewDocumentEnvironment{enumext}{ 0{} }
3670
       \__enumext_safe_exec:
3671
       \__enumext_parse_keys:n {#1}
3672
       \__enumext_before_list:
3673
       \__enumext_start_store_level:
3674
       \__enumext_start_list:nn
         { \tl_use:c { l__enumext_label_ \__enumext_level: _tl } }
3678
            \use:c { __enumext_list_arg_two_ \__enumext_level: : }
3679
            \__enumext_before_keys_exec:
3680
         }
       \__enumext_set_item_width:
3681
       \__enumext_after_args_exec:
3682
3683
3684
     {
       \__enumext_second_part:
3685
```

(End of definition for enumext. This function is documented on page 5.)

As we don't want our check to be executed check-ans by levels but on the complete list, we will take it out of the enumext environment using the "hook" function __enumext_after_env:nn.

©2024 by Pablo González L

```
3687 \__enumext_after_env:nn {enumext}
     {
3689
          _enumext_execute_after_env:
     }
```

13.39 The environment keyans

{ 3692

3691 \cs_new_protected:Nn __enumext_keyans_safe_exec:

The environment keyans also based on lists. The main differences with the enumext environment are the nesting and the way the answers (choice) will be stored and checked, this environment is intended exclusively for "multiple choice questions".

_enumext_keyans_safe_exec:

The keyans environment will only be available if the save-ans key is active and can only be used at the "first level" within the enumext environment. We do not want the environment to be nested, so we will set a maximum at this point. If the conditions are not met, an error message will be returned.

```
\bool_if:NF \l__enumext_store_active_bool
                             3693
                                         \msg_error:nnnn { enumext } { wrong-place }{ keyans }{ save-ans }
                                     \int_incr:N \l__enumext_keyans_level_int
                                    \bool_set_true:N \l__enumext_keyans_env_bool
                             3698
                                    \__enumext_keyans_name_and_start:
                                    % Set false for interfering with enumext nested in keyans (yes, its possible and crayze)
                             3700
                                    \bool_set_false:N \l__enumext_store_active_bool
                                     \int_compare:nNnT { \l__enumext_keyans_level_int } > { 1 }
                                         \msg_error:nn { enumext } { keyans-nested }
                             3704
                                       }
                                    \int_compare:nNnT { \l__enumext_level_int } > { 1 }
                                       {
                                         \msg_error:nn { enumext } { keyans-wrong-level }
                             3708
                                       }
                            (End of definition for \__enumext_keyans_safe_exec:.)
                            Parse \lceil \langle key = val \rangle \rceil for keyans environment.
     enumext kevans parse kevs:n
                             3711 \cs_new_protected:Npn \__enumext_keyans_parse_keys:n #1
                                     \keys_set:nn { enumext / keyans } {#1}
                             3713
                            (\mathit{End}\ of\ definition\ for\ \verb|\__enumext_keyans_parse_keys:n.)
                            Same implementation as the one used in the enumext environment.
 enumext before list v:
\ enumext kevans multicols start:
                             3715 \cs_new_protected:Nn \__enumext_before_list_v:
 \__enumext_keyans_multicols_stop:
\__enumext_second_part_v:
                             3717
                                     \__enumext_vspace_above_v:
                                     \__enumext_before_args_exec_v:
                             3718
                                     \dim_compare:nNnT { \l_enumext_minipage_right_v_dim } > { \c_zero_dim }
                                         \dim_set:Nn \l__enumext_minipage_left_v_dim
                                             \linewidth - \l__enumext_minipage_right_v_dim - \l__enumext_minipage_hsep_v_dim
                             3724
                                         \bool_set_true:N \l__enumext_minipage_active_v_bool
                                         \int_gincr:N \g__enumext_minipage_stat_int
                                         \__enumext_keyans_minipage_add_space:
                                         \__enumext_mini_page{ \l__enumext_minipage_left_v_dim }
                             3728
                                       }
                             3730
                                     \__enumext_keyans_multicols_start:
                                  }
                                \cs_new_protected:Nn \__enumext_keyans_multicols_start:
                             3732
                                     \int_compare:nNnT { \l__enumext_columns_v_int } > { 1 }
                                         \dim_compare:nNnT { \l__enumext_columns_sep_v_dim } = { \c_zero_dim }
                                             \dim_set:Nn \l__enumext_columns_sep_v_dim
                             3738
                                                {
                             ©2024 by Pablo González L
```

102 / 156

```
\l__enumext_labelwidth_v_dim + \l__enumext_labelsep_v_dim
                    ) / \l__enumext_columns_v_int
                   - \l__enumext_listoffset_v_dim
              }
            \dim_set_eq:NN \columnsep \l__enumext_columns_sep_v_dim
            \dim_zero:N \columnseprule % no rule here
            \bool_if:NF \l__enumext_minipage_active_v_bool
                \skip_zero:N \multicolsep
                \__enumext_keyans_multi_addvspace:
            \raggedcolumns
            \begin{multicols}{ \l__enumext_columns_v_int }
3756
   \cs_new_protected:Nn \__enumext_keyans_multicols_stop:
3758
       \int_compare:nNnTF { \l__enumext_columns_v_int } > { 1 }
3759
3760
            \__enumext_stop_list:
3761
            \end{multicols}
            \__enumext_unskip_unkern:
            \__enumext_unskip_unkern:
            \par\addvspace{ \l__enumext_multicols_below_v_skip }
         }
3766
3767
            \__enumext_stop_list:
3768
3769
3770
   \cs_new_protected:Nn \__enumext_second_part_v:
3771
3772
       \bool_if:NTF \l__enumext_minipage_active_v_bool
3774
            \int_compare:nNnT { \g__enumext_minipage_stat_int } = { 1 }
              {
                \msg_warning:nn { enumext } { missing-miniright }
                \miniright
3778
            \int_gzero:N \g__enumext_minipage_stat_int
            \__enumext_unskip_unkern: % remove \topsep + [\partopsep]
            \end__enumext_mini_page
            \par\addvspace{ \l__enumext_minipage_after_skip }
         }
         {
              _enumext_keyans_multicols_stop:
3786
       \bool_set_false:N \l__enumext_keyans_env_bool
3788
       \__enumext_after_stop_list_v:
3789
        \__enumext_vspace_below_v:
3790
3791
(End of definition for \_\_enumext_before_list_v: and others.)
```

__enumext_keyans_set_item_width:

The function __enumext_keyans_set_item_width: will set the value of \itemwidth taking into account the value established by the list-offset key.

 $(\textit{End of definition for } \c\c\c) = \texttt{enumext_keyans_set_item_width:.})$

©2024 by Pablo González L

keyans Now we define the environment keyans also based on lists.

```
3803 \NewDocumentEnvironment{keyans}{ O{}} }
3804
       \__enumext_keyans_safe_exec:
3805
       \__enumext_keyans_parse_keys:n {#1}
3806
       \__enumext_before_list_v:
3807
       \__enumext_start_list:nn
3808
         { \tl_use:N \l__enumext_label_v_tl }
            \__enumext_list_arg_two_v:
            \__enumext_before_keys_exec_v:
         }
       \__enumext_keyans_set_item_width:
       \__enumext_after_args_exec_v:
    }
3816
     {
3817
       \__enumext_check_starred_cmd:n { item }
3818
       \__enumext_second_part_v:
3819
3820
```

(End of definition for keyans. This function is documented on page 15.)

13.40 Tagging PDF support for non-standart list environments

The ETeX release 2022-06-01 brings automatic support for tagged PDF in several aspects, including the standard list environments and the list environment. Unfortunately non-standard list environments like keyanspic or the horizontal list environments enumext* and keyans* are not structured in a nice way, i.e. the expected result in the PDF file is the expected one, but the underlying structure is not correct. In simple terms, for tagged PDF a list environment is a list environment, no matter what it looks like in the PDF file.

To maintain a correct list structure when \DocumentMetadata is active, it is necessary to do some things manually. This implementation is an adaptation of my answer thanks to Ulrike Fischer's comments in How can I modify my \item redefinition to be compatible with tagging-pdf.

13.40.1 Socket for tagging support in enumext* and keyans*

We will first define the necessary sockets and their behavior for enumext* and keyans*.

```
start-list-tags
stop-start-tags
stop-list-tags
\__enumext_start_list_tag:
\__enumext_stop_start_list_tag:
\__enumext_stop_list_tag:n
```

```
3821 \socket_new:nn {tagsupport/enumext/starred}{ 1 }
3822 \socket_new_plug:nnn {tagsupport/enumext/starred} {start-list-tags}
    {
3823
        \tag_resume:n {#1}
        \tag_struct_begin:n {tag=LI}
        \tag_struct_begin:n {tag=Lbl}
3827
       \tag_mc_begin:n {tag=Lbl}
    }
3828
3829 \socket_new_plug:nnn {tagsupport/enumext/starred} {stop-start-tags}
3830
        \tag_mc_end:
3831
       \tag_struct_end:n {tag=Lbl}
3832
       \tag_struct_begin:n {tag=LBody}
3833
       \tag_struct_begin:n {tag=text-unit}
3834
       \tag_struct_begin:n {tag=text}
3835
3836
3837 \socket_new_plug:nnn {tagsupport/enumext/starred} {stop-list-tags}
3838
        \tag_struct_end:n {tag=text}
3839
        \tag_struct_end:n {tag=text-unit}
3840
        \tag_struct_end:n {tag=LBody}
3841
        \tag_struct_end:n {tag=LI}
3842
        \tag_suspend:n {#1}
3843
3844
```

And now we'll wrap them so that they're only active when \DocumentMetadata is present.

©2024 by Pablo González L 104/156

```
\IfDocumentMetadataTF
3856
         {
           \socket_assign_plug:nn {tagsupport/enumext/starred} {stop-start-tags}
3857
           \socket_use:nn {tagsupport/enumext/starred} { }
3858
3859
      }
3860
   \cs_new_protected_nopar:Npn \__enumext_stop_list_tag:n #1
3861
3862
       \IfDocumentMetadataTF
3863
           \socket_assign_plug:nn {tagsupport/enumext/starred} {stop-list-tags}
           \socket_use:nn {tagsupport/enumext/starred} {#1}
3867
         } {}
      }
3868
```

(End of definition for start-list-tags and others.)

13.40.2 Socket for tagging support in keyanspic

We will first define the necessary sockets and their behavior for keyanspic environment.

```
start-list-tags We
stop-start-tags 3869
stop-list-tags 3870
\__enumext_anspic_start_list_tag: 3871
\__enumext_anspic_stop_start_list_tag: 3872
\__enumext_anspic_stop_list_tag: 3873
```

```
3869 \socket_new:nn {tagsupport/enumext/keyanspic}{ 0 }
3870 \socket_new_plug:nnn {tagsupport/enumext/keyanspic} {start-list-tags}
    {
       \tag_resume:n {keyanspic}
       \tag_struct_begin:n {tag=LI}
       \tag_struct_begin:n {tag=Lbl}
3874
       \tag_mc_begin:n {tag=Lbl}
3875
     }
3877 \socket_new_plug:nnn {tagsupport/enumext/keyanspic} {stop-start-tags}
3878
       \tag_mc_end:
3879
       \tag_struct_end:n {tag=Lbl}
3880
       \tag_struct_begin:n {tag=LBody}
3881
       \tag_struct_begin:n {tag=text-unit}
3882
       \tag_struct_begin:n {tag=text}
3883
3884
       \tag_mc_begin:n {tag=text}
    }
3885
$ \socket_new_plug:nnn {tagsupport/enumext/keyanspic} {stop-list-tags}
     {
3888
       \tag_mc_end:
       \tag_struct_end:n {tag=text-unit}
       \tag_struct_end:n {tag=text}
       \tag_struct_end:n {tag=LBody}
3891
       \tag_struct_end:n {tag=LI}
3892
       \tag_suspend:n {keyanspic}
3893
3894
```

And now we'll wrap them so that they're only active when \DocumentMetadata is present.

```
3895 \cs_new_protected_nopar:Nn \__enumext_anspic_start_list_tag:
     {
3896
       \IfDocumentMetadataTF
3897
3898
            \socket_assign_plug:nn {tagsupport/enumext/keyanspic} {start-list-tags}
           \socket_use:n {tagsupport/enumext/keyanspic}
      }
3902
3903 \cs_new_protected_nopar:Nn \__enumext_anspic_stop_start_list_tag:
3904
       \IfDocumentMetadataTF
3905
3906
            \socket_assign_plug:nn {tagsupport/enumext/keyanspic} {stop-start-tags}
3907
           \socket_use:nn {tagsupport/enumext/keyanspic}
3908
3909
      }
3910
3911 \cs_new_protected_nopar:Nn \__enumext_anspic_stop_list_tag:
       \IfDocumentMetadataTF
3913
            \socket_assign_plug:nn {tagsupport/enumext/keyanspic} {stop-list-tags}
3915
           \socket_use:nn {tagsupport/enumext/keyanspic}
3916
         } {}
3917
3918
```

©2024 by Pablo González L 105/156

(End of definition for start-list-tags and others.)

13.41 The environment keyanspic and \anspic

The keyanspic environment is a list based environment that uses the same configuration for "spacing" and $\langle label \rangle$ as the keyans environment, but it does not use \item. The $\langle contents \rangle$ are passed to the environment by means of the \anspic command as replacement for \item command and placed inside minipage environments, with the $\langle label \rangle$ centered "above" or "below", adjusting widths and position according to the options passed to the environment.

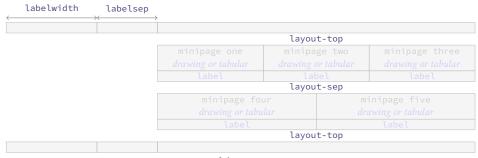


Figure 12: Representation of the keyanspic spacing in enumext.

In order for the keyanspic environment and the \anspic command to work correctly, we need to set and export some variables in the first part of the environment definition and pass them to \anspic which is executed in the second part of the environment. This implementation is adapted from the answer given by Enrico Gregorio (@egreg) in How to process the body of an environment and divide it by a \macro?.

13.41.1 The environment keyanspic

label-pos
label-sep

layout-sty

layout-sep layout-top First we define the key that allows us to process the position of the $\langle label \rangle$ centered "above" or "below" which will be label-pos, the vertical separation of these from drawing or tabular will be handled with the key label-sep. The "layout style" will be handled with the key layout-sty will take two values separated by comma $\{\langle n^\circ upper, n^\circ lower \rangle\}$ and will determine the number of minipage environments in which all arguments of \anspic will be printed at the "upper" and "lower" within the environments separated by the value of the key layout-sep. The vertical space "top" and "bottom" of the environment will be handled with the key layout-top.

```
3919 \keys_define:nn { enumext / keyanspic }
3920
       label-pos .choice:,
3921
       label-pos / above
                            .code:n =
3922
                                  \bool_set_true:N \l__enumext_anspic_label_above_bool
3923
                                  \str_set:Nn \l__enumext_anspic_mini_pos_str { t },
                            .code:n =
       label-pos / below
                                  \bool_set_false:N \l__enumext_anspic_label_above_bool
                                  \str_set:Nn \l__enumext_anspic_mini_pos_str { b },
       label-pos / unknown .code:n =
                                  \msg_error:nneee { enumext } { unknown-choice }
                                    { label-pos } { above,~ below } { \exp_not:n {#1} },
       label-pos
                  .initial:n
                                     = below,
3931
       label-pos
                  .value_required:n = true,
3932
                                     = \l__enumext_anspic_label_sep_skip,
       label-sep
                  .skip set:N
3933
       label-sep .value_required:n = true,
3934
       layout-sty .tl_set:N
                                     = \l__enumext_anspic_layout_style_tl,
       layout-sty .value_required:n = true,
                                     = \keys_set:nn { enumext / keyans }
       layout-sep .code:n
                                          { parsep = #1 },
       lavout-sep .value required:n = true.
       layout-top .code:n
                                     = \keys_set:nn { enumext / keyans }
                                          { topsep = #1 },
3941
       layout-top .value_required:n = true,
3942
                   .code:n
                                     = { \__enumext_keyans_unknown_keys:n {#1} }
       unknown
3943
3944
```

(End of definition for label-pos and others.)

_enumext_keyans_pic_safe_exec: The function __enumext_keyans_pic_safe_exec: check the nested level position inside the enumext environment.

_enumext_keyans_pic_skip_abs:N _enumext_keyans_pic_skip_abs:N _enumext_keyans_pic_arg_two:

_enumext_keyans_pic_arg_two:

_int_incr:N \l_enumext_keyans_pic_level_int \, int_compare:nNnT { \l_enumext_keyans_pic_level_int } > { 1 }

3949 { ©2024 by Pablo González L

106 / 156

The function __enumext_keyans_pic_skip_abs:N will return a positive value \parsep from keyans environment.

The __enumext_keyans_pic_arg_two: function will be used in the *second argument* of the list environment that defines the keyanspic environment, with this we will take the configuration of the "spaces" and the keys label, wrap-label, parsep and topsep from the keyans environment. The first thing we need to do is set the boolean variable \l__enumext_leftmargin_tmp_v_bool handled by the list-indent key to "false", then copy the definition of the second list argument from the keyans environment definition and make sure that \parsep does not have a negative value.

Now we increment the counter enumXv of the keyans environment and save the *total height* of the $\langle label \rangle$ in $\l_enumext_anspic_label_htdp_dim$ used by $\l_enumext_anspic_label_htdp_dim$ used b

```
\bool_if:NF \l__enumext_anspic_label_above_bool
         {
3974
           \stepcounter { enumXv }
3975
           \hbox_set:Nn \l__enumext_anspic_label_box { \l__enumext_label_v_tl }
           \dim_set:Nn \l__enumext_anspic_label_htdp_dim
               \box_ht_plus_dp:N \l__enumext_anspic_label_box
             }
           \skip_add:Nn \parsep
3981
             {
               \l__enumext_anspic_label_htdp_dim
               + \box_dp:N \strutbox
               + \l__enumext_anspic_label_sep_skip
         }
```

Finally we *adjust* the value of \leftmargin and \topsep then set \listparindent, \partopsep and \itemsep to zero so that the *horizontal* and *vertical* space is not affected.

```
3988     \dim_add:Nn \leftmargin { -\labelwidth - \labelsep }
3989     \skip_add:Nn \topsep { 0.5\box_dp:N \strutbox }
3990     \dim_zero:N \listparindent
3991     \skip_zero:N \partopsep
3992     \skip_zero:N \itemsep
3993   }
```

(End of definition for $__$ enumext_keyans_pic_safe_exec: and others.)

keyanspic Now we define the environment keyanspic. For compatibility with *tagged* PDF we must use the \begin{list} form and a lot of conditional code using \IfDocumentMetadataTF.

```
3994 \NewDocumentEnvironment{keyanspic}{ o }
3995 {
3996 \__enumext_keyans_pic_safe_exec:
3997 \__enumext_keyans_pic_parse_keys:n {#1}
```

©2024 by Pablo González L

```
\begin{list} { } { \__enumext_keyans_pic_arg_two: }
       \IfDocumentMetadataTF
            \tag_suspend:n {list}
         }{}
       \item[] \scan_stop:
4003
       % paranoia
       \RenewDocumentCommand \item {}
            \msg_error:nn { enumext } { keyanspic-item-cmd }
         }
       \IfDocumentMetadataTF
         {
4010
            \tag_resume:n {keyanspic}
4011
            \tag_tool:n {para/tagging=false}
4012
            \tag_suspend:n {keyanspic}
4013
         } { }
4014
4015
4016
       \IfDocumentMetadataTF
4017
            \tag_resume:n {keyanspic}
            \tag_struct_begin:n {tag=L,attribute=enumerate}
4021
         } { }
```

Now we process the command \anspic, if the key layout-sty is not present, the number of times the \anspic command appears will be counted from \l_enumext_anspic_args_seq and placed a *single line*.

Finally we check if \anspic* has been used, set the counter enumXvi to zero and apply our "adjusted" vertical space bottom.

```
\__enumext_check_starred_cmd:n { anspic }
       \setcounter { enumXvi } { 0 }
       \bool_if:NTF \l__enumext_anspic_label_above_bool
4035
         {
4036
            \par\addvspace{ 0.5\box_dp:N \strutbox }
4037
         }
4038
         {
4039
            \par
            \addvspace
4041
              {
                \dim_eval:n
                  {
                    \l__enumext_anspic_label_htdp_dim + \box_ht_plus_dp:N \strutbox
                     + \l__enumext_anspic_label_sep_skip + \l__enumext_topsep_v_skip
                  }
              }
4048
         }
4049
4050
```

(End of definition for keyanspic. This function is documented on page 16.)

13.41.2 The command \anspic

The \anspic command take three arguments, the *starred versions* \anspic*[$\langle content \rangle$] *store* the current $\langle label \rangle$ next to the *optional argument* [$\langle content \rangle$] in the *sequence* and *prop list* defined by save-ans key. The third *mandatory argument* { $\langle drawing\ or\ tabular \rangle$ } is NOT *stored* in the *sequence* or *prop list*.

• One of the complications here to make the keyanspic environment compatible with tagged PDF is the position of $\langle label \rangle$, the \anspic command processes the arguments in order, where #1 and #2 correspond to $\langle label \rangle$ and #3 to the mandatory argument and puts all this inside a minipage environment. If #1 and #2, that is $\langle label \rangle$, is above #3 there are no problems with tagged PDF, but if #3 comes first the list created with tagged PDF will not be correct.

\anspic

__enumext_anspic_body_dim:n
__enumext_anspic_label:nn
__enumext_anspic_label_pos:nnn
__enumext_anspic_args:nnn
__enumext_anspic_print:n
__enumext_anspic_print:v
__enumext_anspic_row:n
__enumext_anspic_row:n
__enumext_anspic_exec:

We check that the command is active in the keyanspic environment only if the save-ans key is present, otherwise we return an error. The three arguments are handled by the function __enumext_anspic_args:nnn and stored in the sequence \l__enumext_anspic_args_seq which is processed by the keyanspic environment

```
4051 \NewDocumentCommand \anspic { s o +m }
     {
4052
       \bool_if:NF \l__enumext_store_active_bool
4053
4054
            \msg_error:nnnn { enumext } { wrong-place }{ keyanspic }{ save-ans }
       \int_compare:nNnT { \l__enumext_level_int } > { 1 }
4057
            \msg_error:nn { enumext } { keyanspic-wrong-level }
          }
       \int_compare:nNnT { \l__enumext_keyans_level_int } = { 1 }
4061
4062
            \msg_error:nnnn { enumext } { command-wrong-place }{ anspic }{ keyans }
4063
          }
4064
       \seq_put_right:Nn \l__enumext_anspic_args_seq
4065
              _enumext_anspic_args:nnn { #1 } { #2 } { #3 }
```

The __enumext_anspic_body_dim:n function will set the value of \l__enumext_anspic_body_htdp_-dim equal to the "height plus depth" of the *mandatory argument* if the key label-pos is set "below".

```
4070 \cs_new_protected:Npn \__enumext_anspic_body_dim:n #1
     {
4071
       \bool_if:NF \l__enumext_anspic_label_above_bool
4072
4073
           \IfDocumentMetadataTF
4074
4075
                \tag_suspend:n {keyanspic}
             } { }
           \vbox_set:Nn \l__enumext_anspic_body_box { #1 }
           \dim_set:Nn \l__enumext_anspic_body_htdp_dim
                \box_ht_plus_dp:N \l__enumext_anspic_body_box
4081
4082
           \IfDocumentMetadataTF
                \tag_resume:n {keyanspic}
             } { }
         }
4088
```

The __enumext_anspic_label:nn function will process inside \makebox the starred argument '*' and optional argument passed to the command. Here we will store the $\langle label \rangle$ and optional argument in prop list and sequence and execute the show-ans, show-pos, font, wrap-label and wrap-opt keys.

```
4089 \cs_new_protected:Npn \__enumext_anspic_label:nn #1 #2
       \makebox[ \l__enumext_anspic_mini_width_dim ][ c ]
4091
4092
           \bool_if:nT { #1 }
               \__enumext_keyans_addto_prop:n { #2 }
               \__enumext_keyans_store_ref:
               \__enumext_keyans_addto_seq:n { #2 }
               \int_gincr:N \g__enumext_check_starred_cmd_int
               \bool_lazy_or:nnT
                 { \bool_if_p:N \l__enumext_show_answer_bool }
                 { \bool_if_p:N \l__enumext_show_position_bool }
                 {
                   \tl_set_eq:NN \l__enumext_label_v_tl \l__enumext_label_vi_tl
                   \__enumext_keyans_show_left:n { #2 }
                   \tl_set_eq:NN \l__enumext_label_vi_tl \l__enumext_label_v_tl
                 }
           \tl_use:N \l__enumext_label_font_style_v_tl
           \__enumext_wrapper_label_v:n { \l__enumext_label_vi_tl }
           \__enumext_keyans_show_item_opt:
```

```
)
)
)
```

The function __enumext_anspic_label_pos:nnn will be in charge of handling the "counter" and the position of the $\langle label \rangle$, set by label-pos key which will have the same configuration as the keyans environment.

```
\cs_new_protected:Npn \__enumext_anspic_label_pos:nnn #1 #2 #3
4114
      \stepcounter { enumXvi }
4115
       \__enumext_anspic_body_dim:n { #3 }
      \bool_if:NTF \l__enumext_anspic_label_above_bool
            }
        {
          \raisebox
            {
              -\dim_eval:n
                {
                  \l__enumext_anspic_label_htdp_dim
                  + \l__enumext_anspic_body_htdp_dim
                  + \box_dp:N \strutbox
                  + \l__enumext_anspic_label_sep_skip
                }
            }
            [ Opt ] [ Opt ]
            {
              \__enumext_anspic_label:nn { #1 } { #2 }
4134
        }
```

The $_$ enumext_anspic_args:nnn function will be responsible for placing the code compatible with tagged PDF and the arguments within the $_$ enumext_anspic_args_seq sequence which will be processed by the $_$ enumext_anspic_print:n function in the second part of the definition of the keyanspic environment.

The value $\{\langle n^o upper, n^o lower \rangle\}$ passed to the layout-sty key is split by comma and is handled directly by the function __enumext_anspic_print:n and passed to the function __enumext_anspic_row:n.

```
4153 \cs_new_protected:Nn \__enumext_anspic_print:n
4154 {
4155 \clist_map_function:nN { #1 } \__enumext_anspic_row:n
4156 }
4157 \cs_generate_variant:Nn \__enumext_anspic_print:n { e, V }
```

The function __enumext_anspic_row:n will set the *widths* for the minipage environments and place *all* arguments passed to \anspic saved in the \l__enumext_anspic_args_seq sequence inside them.

```
} { }
           \begin{minipage}[ \l__enumext_anspic_mini_pos_str ]{ \l__enumext_anspic_mini_width_dim }
             \centering
             \seq_item:Nn \l__enumext_anspic_args_seq { ##1 }
           \end{minipage}
           \IfDocumentMetadataTF
4176
               \tag_resume:n {minipage}
             } { }
         }
       \par
4181
```

_enumext_anspic_exec: function will execute all the code in the \anspic command in the second argument of the keyanspic environment definition. If the key layout-sty is not set, everything will be printed on a single line.

```
4182 \cs_new_protected:Nn \__enumext_anspic_exec:
4183
       \tl_if_empty:NTF \l__enumext_anspic_layout_style_tl
4184
         {
              _enumext_anspic_print:e { \seq_count:N \l__enumext_anspic_args_seq }
4186
         }
         {
              _enumext_anspic_print:V \l__enumext_anspic_layout_style_tl
         }
```

(End of definition for \anspic and others. This function is documented on page 16.)

13.42 The horizontal environments

Generating horizontal list environments is NOT as simple as standard LTFX list environments. The fundamental part of the code is adapted from the shortlst package to a more modern version using expl3. It is not possible to redefine \item and \makelabel using \RenewDocumentCommand as in the vertical non starred versions.

To achieve the *horizontal list environments* we will capture the *\item* command and the *\content\rangle* of this in horizontal box using \makebox for the label and a minipage environment for the \(\content \rangle \) passed to \item, we will also add the *optional argument* (\(\lambda u u ber \)) to \(\text{item}\) to be able to join columns horizontally, in simple terms, we want \item to behave in the same way as in the enumext environment but adding an first optional $argument (\langle number \rangle).$

A side effect is the limitation of using \item in this way without using \RenewDocumentCommand, which loses the original definition and affects the standard list environments provided by LTEX and any environment defined using base list environment, including: itemize, enumerate, description, quote, quotation, verse, center, flushleft, flushright, verbatim, tabbing, trivlist, list and all environments created with \newtheorem.

One way to get around this is to use something like:

\AddToHook{env/enumerate/before}{recover original \item definition}

inside minipage, but in my partial tests this does not have the desired effect and the vertical and horizontal spacing is distorted. For now this will remain as a limitation and I will see if it is feasible to implement it in the future.

💎 For compatibility with the tagged PDF we close the environments according to the presence or not of the mini-env key.

13.42.1 Functions for item box width

__enumext_starred_columns_set_vii: \ enumext starred columns set viii: We set the default value for the width of the box containing the $\langle content \rangle$ of the items for enumext* environment.

```
\cs_new_protected:Nn \__enumext_starred_columns_set_vii:
4193
       \dim_compare:nNnT { \l__enumext_columns_sep_vii_dim } = { \c_zero_dim }
           \dim_set:Nn \l__enumext_columns_sep_vii_dim
               ( \l__enumext_labelwidth_vii_dim + \l__enumext_labelsep_vii_dim )
               / \l__enumext_columns_vii_int
4199
             }
       \int_set:Nn \l__enumext_tmpa_vii_int { \l__enumext_columns_vii_int - 1 }
       \dim_set:Nn \l__enumext_item_width_vii_dim
         {
             \linewidth - \l__enumext_columns_sep_vii_dim * \l__enumext_tmpa_vii_int )
             \l__enumext_columns_vii_int
           - \l__enumext_labelwidth_vii_dim
©2024 by Pablo González L
```

```
\l__enumext_labelsep_vii_dim
         }
When the key rightmargin is active we must adjust the values.
       \dim_compare:nNnT { \l__enumext_rightmargin_vii_dim } > { \c_zero_dim }
           \dim_sub:Nn \l__enumext_item_width_vii_dim
             {
               ( \l__enumext_rightmargin_vii_dim * \l__enumext_tmpa_vii_int )
               / \l__enumext_columns_vii_int
4216
           \dim_add:Nn \l__enumext_columns_sep_vii_dim
4217
             {
4218
               \l__enumext_rightmargin_vii_dim
4219
         }
Same implementation for the keyans* environment.
   \cs_new_protected:Nn \__enumext_starred_columns_set_viii:
       \dim_compare:nNnT { \l__enumext_columns_sep_viii_dim } = { \c_zero_dim }
           \dim_set:Nn \l__enumext_columns_sep_viii_dim
             {
               ( \l__enumext_labelwidth_viii_dim + \l__enumext_labelsep_viii_dim )
                / \l__enumext_columns_viii_int
4231
4232
       \int_set:Nn \l__enumext_tmpa_viii_int { \l__enumext_columns_viii_int - 1 }
       \dim_set:Nn \l__enumext_item_width_viii_dim
4234
4235
           ( \linewidth - \l__enumext_columns_sep_viii_dim * \l__enumext_tmpa_viii_int )
4236
           / \l__enumext_columns_viii_int
           - \l__enumext_labelwidth_viii_dim
           - \l__enumext_labelsep_viii_dim
       \dim_compare:nNnT { \l__enumext_rightmargin_viii_dim } > { \c_zero_dim }
4241
         {
           \dim_sub:Nn \l__enumext_item_width_viii_dim
             {
                ( \l__enumext_rightmargin_viii_dim * \l__enumext_tmpa_vii_int )
                / \l__enumext_columns_viii_int
           \dim_add:Nn \l__enumext_columns_sep_viii_dim
             {
                \l__enumext_rightmargin_viii_dim
         }
4253
```

(End of definition for __enumext_starred_columns_set_vii: and __enumext_starred_columns_set_viii:.)

13.42.2 Functions for join item columns

__enumext_starred_joined_item_vii:n
__enumext_starred_joined_item_viii:n

The functions __enumext_starred_joined_item_vii:n and __enumext_starred_joined_item_viii:n will set the width of the box in which the $\langle content \rangle$ passed to $\langle columns \rangle$ will be stored together with the value of $\langle columns \rangle$ environment.

```
4254 \cs_new_protected:Npn \__enumext_starred_joined_item_vii:n #1
       \int_set:Nn \l__enumext_joined_item_vii_int {#1}
4256
       \int_compare:nNnT { \l__enumext_joined_item_vii_int } > { \l__enumext_columns_vii_int }
4258
           \msg_warning:nnee { enumext } { item-joined }
             { \int_use:N \l__enumext_joined_item_vii_int }
             { \int_use:N \l__enumext_columns_vii_int }
           \int_set:Nn \l__enumext_joined_item_vii_int
                \l__enumext_columns_vii_int - \l__enumext_item_column_pos_vii_int + 1
             }
         }
       \int compare:nNnT
         { \l__enumext_joined_item_vii_int }
©2024 by Pablo González L
                                                                                                112 / 156
```

```
{ \l__enumext_columns_vii_int - \l__enumext_item_column_pos_vii_int + 1 }
           \msg_warning:nnee { enumext } { item-joined-columns }
             { \int_use:N \l__enumext_joined_item_vii_int }
               \int eval:n
                 { \l__enumext_columns_vii_int - \l__enumext_item_column_pos_vii_int + 1 }
           \int_set:Nn \l__enumext_joined_item_vii_int
             {
                  __enumext_columns_vii_int - \l__enumext_item_column_pos_vii_int + 1
       \int_compare:nNnTF { \l__enumext_joined_item_vii_int } > { 1 }
4283
         {
4284
           \int_set_eq:NN \l__enumext_joined_item_aux_vii_int \l__enumext_joined_item_vii_int
4285
           \int_decr:N \l__enumext_joined_item_aux_vii_int
4286
           \int_add:Nn \l__enumext_item_column_pos_vii_int { \l__enumext_joined_item_aux_vii_int }
4287
           \int_gadd:Nn \g__enumext_item_count_all_vii_int { \l__enumext_joined_item_aux_vii_int }
           \dim_set:Nn \l__enumext_joined_width_vii_dim
             {
               \l__enumext_item_width_vii_dim * \l__enumext_joined_item_vii_int
                   \l__enumext_labelwidth_vii_dim + \l__enumext_labelsep_vii_dim
                  + \l__enumext_columns_sep_vii_dim
                 )*\l__enumext_joined_item_aux_vii_int
           \dim_set_eq:NN \itemwidth \l__enumext_joined_width_vii_dim
         }
4297
           \dim_set_eq:NN \l__enumext_joined_width_vii_dim \l__enumext_item_width_vii_dim
           \dim_set_eq:NN \itemwidth \l__enumext_item_width_vii_dim
Same implementation for the keyans* environment.
   \cs_new_protected:Npn \__enumext_starred_joined_item_viii:n #1
       \int_set:Nn \l__enumext_joined_item_viii_int {#1}
       \int_compare:nNnT { \l__enumext_joined_item_viii_int } > { \l__enumext_columns_viii_int }
         {
           \msg_warning:nnee { enumext } { item-joined }
             { \int_use:N \l__enumext_joined_item_viii_int }
             { \int_use:N \l__enumext_columns_viii_int }
           \int_set:Nn \l__enumext_joined_item_viii_int
4311
4312
               \l__enumext_columns_viii_int - \l__enumext_item_column_pos_viii_int + 1
       \int compare:nNnT
         { \l__enumext_joined_item_viii_int }
         { \l__enumext_columns_viii_int - \l__enumext_item_column_pos_viii_int + 1 }
           \msg_warning:nnee { enumext } { item-joined-columns }
             { \int_use:N \l__enumext_joined_item_viii_int }
               \int_eval:n
                 { \l__enumext_columns_viii_int - \l__enumext_item_column_pos_viii_int + 1 }
           \int_set:Nn \l__enumext_joined_item_viii_int
             {
               \l__enumext_columns_viii_int - \l__enumext_item_column_pos_viii_int + 1
       \int_compare:nNnTF { \l__enumext_joined_item_viii_int } > { 1 }
         {
           \int_set_eq:NN \l__enumext_joined_item_aux_viii_int \l__enumext_joined_item_viii_int
           \int_decr:N \l__enumext_joined_item_aux_viii_int
           \int_add:Nn \l__enumext_item_column_pos_viii_int { \l__enumext_joined_item_aux_viii_int }
           \int_gadd:Nn \g__enumext_item_count_all_viii_int { \l__enumext_joined_item_aux_viii_int }
           \dim_set:Nn \l__enumext_joined_width_viii_dim
```

113 / 156

(End of definition for __enumext_starred_joined_item_vii:n and __enumext_starred_joined_item_viii:n.)

13.42.3 Functions for mini-env, mini-right and mini-right* keys

__enumext_start_mini_vii:
__enumext_stop_mini_vii:

The implementation of the mini-env key support is almost identical to the one used in the enumext and keyans environments, the difference is that the __enumext_mini_page environment on the "right side" is executed "after" closing the environment, so it is necessary to make a global copy of the variable \l_-enumext_minipage_right_vii_dim in the variable \g__enumext_minipage_right_vii_dim.

```
\cs_new_protected:Nn \__enumext_start_mini_vii:
       \dim_compare:nNnT { \l__enumext_minipage_right_vii_dim } > { \c_zero_dim }
           \dim_set:Nn \l__enumext_minipage_left_vii_dim
             {
               \linewidth
               - \l__enumext_minipage_right_vii_dim
               - \l__enumext_minipage_hsep_vii_dim
           \bool_set_true:N \l__enumext_minipage_active_vii_bool
4362
           \dim_gset_eq:NN
4363
             \g__enumext_minipage_right_vii_dim
             \l__enumext_minipage_right_vii_dim
4365
           \__enumext_mini_addvspace_vii:
           \nointerlineskip\noindent
           \__enumext_mini_page{ \l__enumext_minipage_left_vii_dim }
```

The function __enumext_stop_mini_vii: closes the __enumext_mini_page environment on the "left side", applies \hfill and set the variable \g__enumext_minipage_active_vii_bool to "true" which will be used in the function __enumext_after_env:nn to execute the minipage on the "right side". At this point we will execute the __enumext_stop_list: and __enumext_stop_store_level_vii: functions stopping the list environment and the level saving mechanism for storage in sequence of the \anskey command and anskey* environment. This function is passed to the __enumext_after_list_vii: function in the second part of the enumext* environment definition (§13.43).

```
4371 \cs_new_protected:Nn \__enumext_stop_mini_vii:
     {
       \bool_if:NTF \l__enumext_minipage_active_vii_bool
4373
         {
            \__enumext_stop_list:
            \__enumext_stop_store_level_vii:
4376
            \IfDocumentMetadataTF { \tag_resume:n {enumext*} } { }
           \end__enumext_mini_page
            \hfill
            \bool_gset_true:N \g__enumext_minipage_active_vii_bool
         }
4381
         {
4382
              enumext stop list:
4383
            \__enumext_stop_store_level_vii:
4384
         }
4385
      }
4386
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_start_mini_vii:\ and\ \verb|_-enumext_stop_mini_vii:.)$

Finally we execute the $\{\langle code \rangle\}$ passed to the mini-right or mini-right* keys stored in the variable \g_-enumext_miniright_code_vii_tl in the minipage environment on the "right side". For compatibility with the caption package and possibly other $\{\langle code \rangle\}$ passed to this key, we will pass it to a box and then print it.

```
4387 \__enumext_after_env:nn {enumext*}
                                   {
                                      \bool_if:NT \g__enumext_minipage_active_vii_bool
                               4389
                               4390
                                             _enumext_minipage:w [ t ] { \g__enumext_minipage_right_vii_dim }
                               4391
                                            \legacy_if_gset_false:n { @minipage }
                               4392
                                             \skip_vertical:N \c_zero_skip
                                             \par\addvspace { \g__enumext_minipage_right_skip }
                                             \bool_if:NF \g__enumext_minipage_center_vii_bool
                                                 \tl_put_left:Nn \g__enumext_miniright_code_vii_tl
                                                   {
                                                     \centering
                                                   }
                                             \vbox_set_top:Nn \l__enumext_miniright_code_vii_box
                               4402
                                              {
                               4403
                                                 \tl_use:N \g__enumext_miniright_code_vii_tl
                               4404
                               4405
                                             \box_use_drop:N \l__enumext_miniright_code_vii_box
                                            \skip_vertical:N \c_zero_skip
                                           \__enumext_endminipage:
                                          \par\addvspace{ \g__enumext_minipage_after_skip }
                               4410
                                      \bool_gset_false:N \g__enumext_minipage_active_vii_bool
                               4411
                                      \bool_gset_true:N \g__enumext_minipage_center_vii_bool
                               4412
                                      \tl_gclear:N \g__enumext_miniright_code_vii_tl
                               4413
                                      \dim_gzero:N \g__enumext_minipage_right_vii_dim
                               4414
                                      \bool_gset_false:N \g__enumext_starred_bool
                               4415
                                    }
                              The implementation of the mini-env, mini-right and mini-right* keys is identical to the one used in the
\__enumext_start_mini_viii:
                              enumext* environment.
\__enumext_stop_mini_viii:
                               4417 \cs_new_protected:Nn \__enumext_start_mini_viii:
                                    {
                                      \dim_compare:nNnT { \l__enumext_minipage_right_viii_dim } > { \c_zero_dim }
                               4419
                                        {
                               4420
                                          \dim_set:Nn \l__enumext_minipage_left_viii_dim
                               4421
                               4422
                                            {
                                               \linewidth
                               4423
                                               - \l__enumext_minipage_right_viii_dim
                               4424
                                                \l__enumext_minipage_hsep_viii_dim
                               4425
                                          \bool_set_true:N \l__enumext_minipage_active_viii_bool
                                          \dim_gset_eq:NN
                                             \g__enumext_minipage_right_viii_dim
                                            \l__enumext_minipage_right_viii_dim
                                          \__enumext_mini_addvspace_viii:
                               4431
                                          \nointerlineskip\noindent
                               4432
                                          \__enumext_mini_page{ \l__enumext_minipage_left_viii_dim }
                               4433
                                        }
                               4434
                               4435
                               4436 \cs_new_protected:Nn \__enumext_stop_mini_viii:
                               4437
                                      \bool_if:NTF \l__enumext_minipage_active_viii_bool
                                        {
                                          \__enumext_stop_list:
                                          \IfDocumentMetadataTF { \tag_resume:n {keyans*} } { }
                               4441
                                          \end__enumext_mini_page
                               4442
                                          \hfill
                               4443
                                          \bool_gset_true:N \g__enumext_minipage_active_viii_bool
                               4444
                               4445
                                        {
                                           \__enumext_stop_list:
                                        }
                                  \__enumext_after_env:nn {keyans*}
                               4451
                                      \bool_if:NT \g__enumext_minipage_active_viii_bool
                               4453
```

©2024 by Pablo González L

__enumext_mini_page{ \g__enumext_minipage_right_viii_dim }

4454

```
\par\addvspace { \g__enumext_minipage_right_skip }
             \bool_if:NF \g__enumext_minipage_center_viii_bool
                 \tl_put_left:Nn \g__enumext_miniright_code_viii_tl
                   {
                     \centering
             \vbox_set_top:Nn \l__enumext_miniright_code_viii_box
                 \tl_use:N \g__enumext_miniright_code_viii_tl
             \box_use_drop:N \l__enumext_miniright_code_viii_box
           \end__enumext_mini_page
           \par\addvspace{ \g__enumext_minipage_after_skip }
4469
4470
       \bool_gset_false:N \g__enumext_minipage_active_viii_bool
4471
       \bool_gset_true:N \g__enumext_minipage_center_viii_bool
4472
       \tl_gclear:N \g__enumext_miniright_code_viii_tl
4473
       \dim_gzero:N \g__enumext_minipage_right_viii_dim
4474
```

 $(\textit{End of definition for } \c\c\c) = \texttt{enumext_start_mini_viii:} \ \ \textit{and } \c\c\c) = \texttt{enumext_stop_mini_viii:}.)$

13.42.4 Redefining \footnote command

__enumext_footnotetext:nn
__enumext_renew_footnote:
__enumext_print_footnote:

To keep the correct numbering of \footnote and to make it work correctly in the enumext* and keyans* environments, it is necessary to redefine the command. This implementation is adapted from the answer given by Clea F. Rees (@cfr) in footnotes in boxes compatible with hyperref.

```
4476 \cs_new_protected:Nn \__enumext_footnotetext:nn
    {
4477
      \footnotetext[#1]{#2}
4478
4479
  \verb|\cs_new_protected:Nn \  | \_enumext_renew_footnote: \\
4481
       \seq_gclear:N \g__enumext_footnote_arg_seq
4482
       \seq_gclear:N \g__enumext_footnote_int_seq
       \RenewDocumentCommand \footnote { o +m }
           \tl_if_novalue:nTF {##1}
4486
            {
4487
              \stepcounter{footnote}
4488
              \int_gset_eq:Nc \g__enumext_footnote_int { c@footnote }
             {
              \int_gset:Nn \g__enumext_footnote_int { ##1 }
           \footnotemark [ \g__enumext_footnote_int ]
           \seq_gput_right:Nn \g__enumext_footnote_arg_seq { ##2 }
           }
4497
4498
  \cs_new_protected:Nn \__enumext_print_footnote:
4499
4500
       \seq_if_empty:NF \g__enumext_footnote_int_seq
4501
           \seq_map_pairwise_function:NNN
             \g__enumext_footnote_int_seq
             \g__enumext_footnote_arg_seq
             \__enumext_footnotetext:nn
        }
4507
    }
4508
```

 $(\textit{End of definition for } \\ _\texttt{enumext_footnoteext:nn,} \\ _\texttt{enumext_renew_footnote:.})$

13.43 The environment enumext*

enumext* First we will generate the environment and we will give a temporary definition to __enumext_stop_-item_tmp_vii: equal to __enumext_first_item_tmp_vii: and next to \item equal to __enumext_-start_item_tmp_vii: which we will redefine later. Unlike the implementation used by the shortlst package, we will not set the values of \rightskip and \@rightskip equal to \@flushglue whose value is

0.0pt plus 1.0 fil, in the tests I have performed this fails in some circumstances and different results are obtained when using pdfTeX and LuaTeX.

```
4509 \NewDocumentEnvironment{enumext*}{ o }
4510
       \__enumext_safe_exec_vii:
       \__enumext_parse_keys_vii:n {#1}
4512
       \__enumext_before_list_vii:
       \__enumext_start_store_level_vii:
4514
       \__enumext_start_list:nn { }
4515
4516
            \__enumext_list_arg_two_vii:
4517
            \__enumext_before_keys_exec_vii:
4518
4519
       \IfDocumentMetadataTF { \tag_suspend:n {enumext*} } { }
       \__enumext_starred_columns_set_vii:
       \item[] \scan_stop:
       \cs_set_eq:NN \__enumext_stop_item_tmp_vii: \__enumext_first_item_tmp_vii:
       \cs_set_eq:NN \item \__enumext_start_item_tmp_vii:
       \ignorespaces
     }
4527
       \IfDocumentMetadataTF { \tag_struct_end:n {tag=text-unit} } { }
4528
       \__enumext_stop_item_tmp_vii:
4529
       \__enumext_remove_extra_parsep_vii:
       \__enumext_after_list_vii:
4531
     }
```

(End of definition for enumext*. This function is documented on page 5.)

__enumext_safe_exec_vii:

We will first call the function __enumext_internal_mini_page: to create the environment __enumext_-mini_page, then the function __enumext_is_not_nested: which sets \g__enumext_starred_bool to true if we are not nested within enumext, we will increment \l__enumext_level_h_int to restrict nesting of the environment, set \l__enumext_starred_bool to true and finally call the function __enumext_is_-on_first_level: which sets \l__enumext_starred_first_bool to true if we are not nested, allowing the "storage system" to be used.

```
4533 \cs_new_protected:Nn \__enumext_safe_exec_vii:
4534
       \__enumext_internal_mini_page:
4535
       \__enumext_is_not_nested:
       \int_incr:N \l__enumext_level_h_int
       \int_compare:nNnT { \l__enumext_level_h_int } > { 1 }
           \msg_error:nn { enumext } { nested }
         }
4541
       \int_compare:nNnT { \l__enumext_keyans_level_h_int } = { 1 }
4543
           \msg_error:nnn { enumext } { nested-horizontal } { keyans*}
         }
       \bool_set_true:N \l__enumext_starred_bool
       \bool_set_false:N \l__enumext_standar_bool
       \__enumext_is_on_first_level:
     }
```

(End of definition for __enumext_safe_exec_vii:.)

__enumext_parse_keys_vii:n

First we will clear the variable \l__enumext_series_str used by the key series, process the environment $[\langle key = val \rangle]$ and execute the function __enumext_parse_series:n and used by the key series, then we execute the function __enumext_store_active_keys_vii:n and reprocess the $\langle keys \rangle$ to pass them to the storage *sequence* if the key save-key is not active.

(End of definition for $_$ enumext_parse_keys_vii:n.)

__enumext_before_list_vii:

The function __enumext_before_list_vii: first calls the function __enumext_vspace_above_vii: used by the keys above and above*, then calls the function __enumext_check_ans_active: for the check answer mechanism and finally calls the functions __enumext_before_args_exec: and __enumext_start_mini_vii: used by the keys before*, mini-env, mini-right and mini-right*.

```
4560 \cs_new_protected:Nn \__enumext_before_list_vii:
4561 {
4562 \__enumext_vspace_above_vii:
4563 \__enumext_check_ans_active:
4564 \__enumext_before_args_exec_vii:
4565 \__enumext_start_mini_vii:
4566 }
```

 $(\textit{End of definition for } \verb|__enumext_before_list_vii:.)$

__enumext_after_list_vii:

The function __enumext_after_list_vii: first calls the function __enumext_stop_mini_vii: which internally calls __enumext_stop_list: and __enumext_stop_store_level_vii: (§13.42.3) used by the keys mini-env, mini-right and mini-right*, then to the functions __enumext_after_stop_list_vii: used by the key after, __enumext_check_ans_key_hook: used by the key check-ans, __enumext_vspace_below_vii: used by the keys below and below*. Finally set \l__enumext_starred_bool to false and call the __enumext_resume_save_counter: function used by the series, resume and resume* keys.

```
4567 \cs_new_protected:Nn \__enumext_after_list_vii:
    {
4568
       \__enumext_stop_mini_vii:
       \__enumext_after_stop_list_vii:
4570
       \__enumext_check_ans_key_hook:
4571
       \ enumext vspace below vii:
4572
       \bool_set_false:N \l__enumext_starred_bool
4573
       \__enumext_resume_save_counter:
4574
     }
4575
```

(End of definition for $_$ enumext_after_list_vii:.)

__enumext_start_store_level_vii:
__enumext_stop_store_level_vii:

The __enumext_start_store_level_vii: and __enumext_stop_store_level_vii: functions activate the "storing structure" mechanism in sequence for \anskey command and anskey* environment if enumext* are nested in enumext.

```
4576 \cs_new_protected:Nn \__enumext_start_store_level_vii:
4577
       \bool_if:NT \l__enumext_store_active_bool
4578
4579
           \int_compare:nNnT { \l__enumext_level_int } > { 0 }
                \__enumext_store_level_open_vii:
         }
4584
4585
   \cs_new_protected:Nn \__enumext_stop_store_level_vii:
4586
4587
       \bool_if:NT \l__enumext_store_active_bool
4588
4589
           \int_compare:nNnT { \l__enumext_level_int } > { 0 }
4591
                \__enumext_store_level_close_vii:
         }
```

 $(\textit{End of definition for } _\texttt{enumext_start_store_level_vii:} \ \textit{and } \bot _\texttt{enumext_stop_store_level_vii:})$

13.43.1 The command \item in enumext*

__enumext_first_item_tmp_vii:

The __enumext_first_item_tmp_vii: function will remove horizontal space equal to \labelwidth plus \labelsep to the left of the "first" \item in the environment at the point of execution of this function, where it is equal to the __enumext_stop_item_tmp_vii: function inside the environment body definition.

```
4596 \cs_new_protected_nopar:Nn \__enumext_first_item_tmp_vii:
4597 {
4598 \skip_horizontal:n
4599 {
4600 -\l__enumext_labelwidth_vii_dim - \l__enumext_labelsep_vii_dim
62024 by Pablo González L
```

118 / 156

```
4601      }
4602      \ignorespaces
4603     }

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

__enumext_start_item_tmp_vii:
__enumext_item_peek_args_vii:
__enumext_joined_item_vii:w
__enumext_standar_item_vii:w
__enumext_starred_item_vii_aux_i:w
__enumext_starred_item_vii_aux_ii:w
__enumext_starred_item_vii_aux_iii:w
__enumext_starred_item_vii_aux_iii:w

First we will call the function __enumext_stop_item_tmp_vii: that we will redefine later, we will increment the value of \l__enumext_item_column_pos_vii_int that will count the item's by rows and the value of \g__enumext_item_count_all_vii_int that will count the total of item's in the environment. After that we will call the function __enumext_item_peek_args_vii: that will handle the arguments passed to \item.

```
4604 \cs_new_protected_nopar:Nn \__enumext_start_item_tmp_vii:
4605 {
4606 \__enumext_stop_item_tmp_vii:
4607 \int_incr:N \l__enumext_item_column_pos_vii_int
4608 \int_gincr:N \g__enumext_item_count_all_vii_int
4609 \__enumext_item_peek_args_vii:
4600 }
```

The function __enumext_item_peek_args_vii: will handle the \item($\langle number \rangle$). Look for the argument "(", if it is present we will call the function __enumext_joined_item_vii:w ($\langle number \rangle$), which is in charge of joining the item's in the same row, in case they are not present we will set the default value (1).

The function __enumext_joined_item_vii:w will first call the function __enumext_starred_-joined_item_vii:n in charge of setting the *width* of the box that will store the content passed to \item. Then we will look for the argument "*", if it is present we will call the function __enumext_starred_item_vii:w otherwise we will call the function __enumext_starred_item_vii:w.

```
4617 \cs_new_protected:Npn \__enumext_joined_item_vii:w (#1)
4618 {
4619 \__enumext_starred_joined_item_vii:n {#1}
4620 \peek_meaning_remove:NTF *
4621 { \__enumext_starred_item_vii:w }
4622 { \__enumext_standar_item_vii:w }
4623 }
```

The function __enumext_standar_item_vii:w will first look for the argument "[", if present it will set the state of the variable \l__enumext_wrap_label_opt_vii_bool equal to the state of the variable \l__enumext_wrap_label_opt_vii_bool handled by the key wrap-label* and finally execute the non-enumerated version \item[\langle custom \rangle] by means of the function __enumext_start_item_vii:w, otherwise we will set the value of the variable \l__enumext_wrap_label_vii_bool handled by the wrap-label key to true and set the switch \ifenoitemarg to true to execute the enumerated version of \item by means of the function __enumext_start_item_vii:w [\l__enumext_label_vii_tl].

```
4624 \cs_new_protected:Npn \__enumext_standar_item_vii:w
    {
4625
       \bool_set_false:N \l__enumext_item_starred_vii_bool
4626
       \peek_meaning:NTF [
4627
4628
           \bool_set_eq:NN \l__enumext_wrap_label_vii_bool \l__enumext_wrap_label_opt_vii_bool
           \__enumext_start_item_vii:w
         }
4631
4632
           \bool_set_true:N \l__enumext_wrap_label_vii_bool
4633
           \legacy_if_set_true:n { @noitemarg }
              _enumext_start_item_vii:w [ \l__enumext_label_vii_tl ] \ignorespaces
4637
```

The function __enumext_starred_item_vii:w together with the specified auxiliary functions aux_i:w, aux_ii:w, and aux_iii:w execute \item*, \item*[$\langle symbol \rangle$] and \item*[$\langle symbol \rangle$][$\langle offset \rangle$].

```
4638 \cs_new_protected:Npn \__enumext_starred_item_vii:w
4639 {
4640 \bool_set_true:N \l__enumext_item_starred_vii_bool
4641 \bool_set_true:N \l__enumext_wrap_label_vii_bool
4642 \peek_meaning:NTF [
4643 {\__enumext_starred_item_vii_aux_i:w}
```

```
{ \__enumext_starred_item_vii_aux_ii:w }
     7
  \cs_new_protected:Npn \__enumext_starred_item_vii_aux_i:w [#1]
       \tl_gset:Nn \g__enumext_item_symbol_aux_vii_tl {#1}
4648
       \__enumext_starred_item_vii_aux_ii:w
4649
    }
4650
   \cs_new_protected:Npn \__enumext_starred_item_vii_aux_ii:w
4651
4652
       \peek_meaning:NTF [
4653
         { \__enumext_starred_item_vii_aux_iii:w }
         {
           \dim_set_eq:NN \l__enumext_item_symbol_sep_vii_dim \l__enumext_labelsep_vii_dim
           \legacy_if_set_true:n { @noitemarg }
4657
           \__enumext_start_item_vii:w [ \l__enumext_label_vii_tl ] \ignorespaces
4658
4659
4660
   \cs_new_protected:Npn \__enumext_starred_item_vii_aux_iii:w [#1]
4662
       \dim_set:Nn \l__enumext_item_symbol_sep_vii_dim {#1}
4663
       \legacy_if_set_true:n { @noitemarg }
       \__enumext_start_item_vii:w [ \l__enumext_label_vii_tl ] \ignorespaces
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_start_item_tmp_vii:\ \mathit{and}\ \mathit{others}.)$

\ enumext fake make label vii:n

The __enumext_fake_make_label_vii:n function will be in charge of handling our definition of \item. First we increment the counter enumXvii for the enumerated items and activate support for the *check answers* mechanism, followed by support for \item*[$\langle symbol \rangle$][$\langle offset \rangle$] if present, then the wrap-label and wrap-label* keys which we execute using \makebox whose width will be given by the labelwidth key and position by the align key, inside the argument of this we will execute the font key together with the function defined by the wrap-label or wrap-label* keys. Finally we execute the labelsep key applying a \skip_horizontal:N and \ignorespaces.

◆ For compatibility with tagged PDF and hyperref when an environment enumext is nested in enumext* and the key save-ans is not active need setting the \ifehyper@item switch to "true". The explanation for this is given by the master Heiko Oberdiek on \refstepcounter{enumi} twice (or more) creates destination with the same identifier. This patch is only needed if you are running pdflatex and not if you are running lualatex

```
4667 \cs_new_protected_nopar:Npn \__enumext_fake_make_label_vii:n #1
    {
4668
       \legacy_if:nT { @noitemarg }
4669
           \legacy_if_set_false:n { @noitemarg }
           \legacy_if:nT { @nmbrlist }
             {
               \IfDocumentMetadataTF
                 {
                    \bool_if:NT \l__enumext_hyperref_bool
                      {
                        \legacy_if_set_true:n { @hyper@item }
                 } { }
               \refstepcounter{enumXvii}
               \bool_if:NT \l__enumext_check_answers_bool
                    \int_gincr:N \g__enumext_item_number_int
4684
                    \bool_set_true:N \l__enumext_item_number_bool
                 }
4686
       \bool_if:NT \l__enumext_item_starred_vii_bool
           \tl_if_blank:VT \g__enumext_item_symbol_aux_vii_tl
               \tl_gset_eq:NN
                  \g__enumext_item_symbol_aux_vii_tl \l__enumext_item_symbol_vii_tl
             7
           \mode_leave_vertical:
           \skip_horizontal:n { -\l__enumext_item_symbol_sep_vii_dim }
           \hbox_overlap_left:n { \g__enumext_item_symbol_aux_vii_tl }
           \skip_horizontal:N \l__enumext_item_symbol_sep_vii_dim
©2024 by Pablo González L
```

120 / 156

```
\tl_gclear:N \g__enumext_item_symbol_aux_vii_tl
}

makebox[ \l__enumext_labelwidth_vii_dim ][ \l__enumext_align_label_vii_str ]

{

tl_use:N \l__enumext_label_font_style_vii_tl

bool_if:NTF \l__enumext_wrap_label_vii_bool

{

__enumext_wrapper_label_vii:n {#1}

}

enumext_wrapper_label_vii:n {#1}

}

skip_horizontal:N \l__enumext_labelsep_vii_dim \ignorespaces
}
```

(End of definition for $_=$ enumext_fake_make_label_vii:n.)

13.43.2 Real definition of \item in enumext*

The functions __enumext_start_item_vii:w and __enumext_stop_item_vii: executing the true definition of \item inside the enumext* environment, unlike the implementation in shortlst we will NOT use an extra group and the plain form of the lrbox environment.

__enumext_start_item_vii:w
__enumext_stop_item_vii:

The first thing we will do is set the value of __enumext_stop_item_tmp_vii: equal to __enumext_stop_item_vii: equal to __enumext_stop_item_vii: which we will define later, after that we will start capturing \item and "item content" in a horizontal box where the width will be \itemwidth plus \labelsep.

```
4713 \cs_new_protected_nopar:Npn \__enumext_start_item_vii:w [#1]
4714 {
4715 \cs_set_eq:NN \__enumext_stop_item_tmp_vii: \__enumext_stop_item_vii:
4716 \hbox_set_to_wd:Nnw \l__enumext_item_text_vii_box
4717 {
4718 \l__enumext_joined_width_vii_dim
4719 + \l__enumext_labelwidth_vii_dim
4720 + \l__enumext_labelsep_vii_dim
4721 }
```

If $\lower \$ not active and the state of the variable $\lower \$ not notes_key_bool is false, we will redefine the $\lower \$ not note command.

Now we insert our sockets for tagging PDF support and run \item.

```
4729 \__enumext_start_list_tag:n {enumext*}
4730 \__enumext_fake_make_label_vii:n {#1}
4731 \__enumext_stop_start_list_tag:
```

Finally we open the minipage environment, capture the "item content", make \parindent take the value of the key listparindent and \parskip take the value of the key parsep, then execute the keys itemindent and first.

If Here the use of \unskip and \skip_horizontal:n with the value of listparindent is necessary, otherwise an unwanted space is created when using \item[$\langle opt \rangle$] and the value passed to the key itemindent is incremented.

```
\__enumext_minipage:w [ t ]{ \l__enumext_joined_width_vii_dim }
\dim_set_eq:NN \parindent \l__enumext_listparindent_vii_dim
\skip_set_eq:NN \parskip \l__enumext_parsep_vii_skip
\__enumext_unskip_unkern:
\__enumext_unskip_unkern:
\skip_horizontal:n { -\l__enumext_listparindent_vii_dim } \ignorespaces
\tl_use:N \l__enumext_fake_item_indent_vii_tl
\tl_use:N \l__enumext_after_list_args_vii_tl
\]
```

The __enumext_stop_item_vii: function will finish the fetching \item and "item content" by closing the minipage environment, the sockets for tagging PDF and the horizontal box.

```
4741 \cs_new_protected_nopar:Nn \__enumext_stop_item_vii:
4742 {
4743 \__enumext_endminipage:
4744 \__enumext_stop_list_tag:n {enumext*}
4745 \hbox_set_end:
©2024 by Pablo González L
```

Here we will reduce the *warnings* a bit by setting the value of \hbadness to 10000, print \item and "item content" from the *horizontal box* and *footnotes* if it was redefined.

Finally apply the *vertical space* between rows set by itemsep key passed to \parsep using \par\noindent and *horizontal space* between columns set by columns-sep key using \skip_horizontal:N.

 $(\textit{End of definition for } \c\c\c) = \texttt{enumext_start_item_vii:w} \ \ \textit{and } \c\c\c) = \texttt{enumext_stop_item_vii:.})$

__enumext_remove_extra_parsep_vii:

Remove the extra *vertical space* equal to \parsep=\itemsep when the total number of \item is divisible by the number of \item in the last row of the environment. Here the use of \unskip or \removelastskip fails and does not obtain the expected result, using \vspace is the option and in this case, we can use a simplified version since we are always in \(\frac{\vertical mode}{\circ} \).

```
4765 \cs_new_protected:Nn \__enumext_remove_extra_parsep_vii:
     {
       \int_compare:nNnT
4767
         {
4768
            \int_mod:nn
4769
              { \g__enumext_item_count_all_vii_int } { \l__enumext_columns_vii_int }
         }
4771
         =
4772
          { 0 }
4773
4774
            \para_end:
4775
            \skip_vertical:n { -\l__enumext_itemsep_vii_skip }
            \skip_vertical:N \c_zero_skip
            \int_gzero:N \g__enumext_item_count_all_vii_int
4778
         }
4779
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_remove_extra_parsep_vii:.)$

As we don't want our check to be executed check-ans by levels but on the complete list, we will take it out of the enumext* environment using the "hook" function __enumext_after_env:nn.

```
4781 \__enumext_after_env:nn {enumext*}
4782 {
4783 \__enumext_execute_after_env:
4784 }
```

13.44 The environment keyans *

eyans* The implementation of keyans* environment is the similar as that used by the enumext* environment except for the __enumext_check_starred_cmd:n function added in the second part.

```
\IfDocumentMetadataTF { \tag_suspend:n {keyans*} } { }
       \__enumext_starred_columns_set_viii:
       \item[] \scan_stop:
       \cs_set_eq:NN \__enumext_stop_item_tmp_viii: \__enumext_first_item_tmp_viii:
4798
       \cs_set_eq:NN \item \__enumext_start_item_tmp_viii:
4799
       \ignorespaces
4800
    }
4801
4802
       \IfDocumentMetadataTF { \tag_struct_end:n {tag=text-unit} } { }
4803
       \__enumext_stop_item_tmp_viii:
       \__enumext_remove_extra_parsep_viii:
       \__enumext_check_starred_cmd:n { item }
       \__enumext_after_list_viii:
4807
4808
```

(End of definition for keyans \star . This function is documented on page 15.)

__enumext_safe_exec_viii:

The __enumext_safe_exec_viii: function will first check if the save-ans key is active and only when this is true the environment will be available, it will increment the value of \l__enumext_keyans_level_h_int and return an error message when we are nesting the environment, then it will call the __enumext_-keyans_name_and_start: function in charge of saving the name of the environment and the line it is running on, then it will check if we are trying to nest keyans* in enumext* returning an error and we will set \l__enumext_starred_bool to true, finally we will check if we are within the appropriate level within the enumext environment.

```
4809 \cs_new_protected:Nn \__enumext_safe_exec_viii:
                                  4810
                                          \bool_if:NF \l__enumext_store_active_bool
                                  4811
                                  4812
                                              \msg_error:nnnn { enumext } { wrong-place }{ keyans* }{ save-ans }
                                  4813
                                  4814
                                          \int_incr:N \l__enumext_keyans_level_h_int
                                          \int_compare:nNnT { \l__enumext_keyans_level_h_int } > { 1 }
                                            {
                                  4817
                                              \msg_error:nn { enumext } { nested }
                                  4818
                                            }
                                  4819
                                          \__enumext_keyans_name_and_start:
                                  4820
                                          \bool_if:NT \l__enumext_starred_bool
                                  4821
                                            {
                                  4822
                                              \msg_error:nnn { enumext } { nested-horizontal } { enumext* }
                                  4823
                                  4824
                                          \bool_set_true:N \l__enumext_starred_bool
                                  4825
                                          % Set false for interfering with enumext nested in keyans* (yes, its possible and crayze)
                                          \bool_set_false:N \l__enumext_store_active_bool
                                          \int_compare:nNnT { \l__enumext_level_int } > { 1 }
                                              \msg_error:nn { enumext } { keyans-wrong-level }
                                            }
                                  4831
                                       }
                                  4832
                                 (End of definition for \_=enumext\_safe\_exec\_viii:.)
\__enumext_parse_keys_viii:n
                                 Parse [\langle key = val \rangle] for keyans*.
                                  4833 \cs_new_protected:Npn \__enumext_parse_keys_viii:n #1
                                       {
                                          \tl_if_novalue:nF {#1}
                                  4835
                                  4836
                                            {
                                              \keys_set:nn { enumext / keyans* } {#1}
                                  4837
                                            }
                                  4838
                                  4839
                                 (\mathit{End}\ of\ definition\ for\ \verb|\__enumext_parse_keys_viii:n.)
```

__enumext_before_list_viii:

The function $_$ _enumext_before_list_viii: will add the vertical spacing on the environment if the above key is active next to the $\{\langle code \rangle\}$ defined by the before* key if it is active, the call the function $_$ enumext_start_mini_viii: handle by mini-env.

(End of definition for __enumext_before_list_viii:.)

__enumext_after_list_viii:

The function __enumext_after_list_viii: first call the function __enumext_stop_mini_viii:, then apply the $\{\langle code \rangle\}$ handled by the after key together with the *vertical space* handled by the below key if they are present.

```
4846 \cs_new_protected:Nn \__enumext_after_list_viii:
4847 {
4848 \__enumext_stop_mini_viii:
4849 \__enumext_after_stop_list_viii:
4850 \__enumext_vspace_below_viii:
4851 }
```

(End of definition for $_=$ enumext_after_list_viii:.)

13.44.1 The command \item in keyans*

The idea here is to make the \item command behave in the same way as in the keyans environment with the difference of the *optional argument* ($\langle number \rangle$) which works in the same way as in the enumext* environment. In simple terms we want to store the $\langle label \rangle$ next to the $\lceil \langle content \rangle \rceil$ if it is present in the *sequence* and *prop list* defined by save-ans key for \item*, \item* $\lceil \langle content \rangle \rceil$, \item($\langle number \rangle$)* and \item($\langle number \rangle$) \int $\lceil \langle content \rangle \rceil$ commands.

__enumext_first_item_tmp_viii:

The __enumext_first_item_tmp_viii: function will remove horizontal space equal to \labelwidth plus \labelsep to the left of the "first" \item in the environment at the point of execution of this function, where it is equal to the __enumext_stop_item_tmp_viii: function inside the environment body definition.

(End of definition for __enumext_first_item_tmp_viii:.)

__enumext_start_item_tmp_viii:
__enumext_item_peek_args_viii:
__enumext_joined_item_viii:w
_enumext_standar_item_viii:w

First we will call the function __enumext_stop_item_tmp_viii: that we will redefine later, we will increment the value of \l__enumext_item_column_pos_viii_int that will count the item's by rows and the value of \g__enumext_item_count_all_viii_int that will count the total of item's in the environment. After that we will call the function __enumext_item_peek_args_viii: that will handle the arguments passed to \item.

The function $_$ _enumext_item_peek_args_viii: will handle the \identified (\number). Look for the argument "(", if it is present we will call the function $_$ _enumext_joined_item_viii: w (\number), which is in charge of joining the item's in the same row, in case they are not present we will set the default value (1).

The function __enumext_joined_item_viii:w will first call the function __enumext_starred_-joined_item_viii:n in charge of setting the *width* of the box that will store the content passed to \item. Then we will look for the argument "*", if it is present we will call the function __enumext_starred_-item_viii:w otherwise we will call the function __enumext_standar_item_viii:w.

```
4873 \cs_new_protected:Npn \__enumext_joined_item_viii:w (#1)
4874 {
4875 \__enumext_starred_joined_item_viii:n {#1}
4876 \peek_meaning_remove:NTF *
4877 {\__enumext_starred_item_viii:w }
4878 {\__enumext_standar_item_viii:w }
4879 }
```

The function __enumext_standar_item_viii:w will first look for the argument "[", if present it will set the state of the variable \l__enumext_wrap_label_opt_viii_bool equal to the state of the variable \l__enumext_wrap_label_opt_viii_bool handled by the key wrap-label* and finally execute the non-enumerated version \item[$\langle custom \rangle$] by means of the function __enumext_start_item_viii:w, otherwise we will set the value of the variable \l__enumext_wrap_label_viii_bool handled by the wrap-label key to true and set the switch \if@noitemarg to true to execute the enumerated version of \item by means of the function __enumext_start_item_viii:w [\l__enumext_label_viii_tl].

```
4880 \cs_new_protected:Npn \__enumext_standar_item_viii:w
    {
4881
       \bool_set_false:N \l__enumext_item_starred_viii_bool
4882
       \peek_meaning:NTF [
4883
           \bool_set_eq:NN \l__enumext_wrap_label_viii_bool \l__enumext_wrap_label_opt_viii_bool
           \__enumext_start_item_viii:w
         }
         {
4888
           \bool_set_true:N \l__enumext_wrap_label_viii_bool
4889
           \legacy_if_set_true:n { @noitemarg }
4890
           \__enumext_start_item_viii:w [ \l__enumext_label_viii_tl ] \ignorespaces
4891
    }
4893
```

(End of definition for __enumext_start_item_tmp_viii: and others.)

__enumext_starred_item_viii:w __enumext_starred_item_viii_aux_i:w __enumext_starred_item_viii_aux_ii:w __enumext_starred_item_exec: The function __enumext_starred_item_viii:w together with the specified auxiliary functions aux_i:w and aux_ii:w execute \item* and \item* [$\langle content \rangle$].

The function __enumext_starred_item_viii_aux_i:w will save the *optional argument* to \item* in \l__enumext_store_current_opt_arg_tl and will save this argument along with the spacing set by the key save-sep in variable \l__enumext_store_current_label_tl if present, then call the function __enumext_starred_item_viii_aux_ii:w.

```
4902 \cs_new_protected:Npn \__enumext_starred_item_viii_aux_i:w [#1]
4903
       \tl_clear:N \l__enumext_store_current_label_tl
4904
       \tl_if_novalue:nF { #1 }
4905
           \tl_if_empty:NF \l__enumext_store_keyans_item_opt_sep_tl
               \tl_put_right:Ne \l__enumext_store_current_label_tl
                    \l__enumext_store_keyans_item_opt_sep_tl
                  }
4912
               \tl_put_right:Ne \l__enumext_store_current_label_tl { #1 }
4913
4914
           \tl_set:Ne \l__enumext_store_current_opt_arg_tl { #1 }
4915
4916
       \__enumext_starred_item_viii_aux_ii:w
4917
     }
4918
   \cs_new_protected:Npn \__enumext_starred_item_viii_aux_ii:w
       \legacy_if_set_true:n { @noitemarg }
4921
         _enumext_start_item_viii:w [ \l__enumext_label_viii_tl ] \ignorespaces
```

The function __enumext_starred_item_exec: will be in charge of storing the current $\langle label \rangle$ for \item* followed by the $[\langle content \rangle]$ for \item* $[\langle content \rangle]$ if present in the sequence and prop list set by the save-ans key. In this same function the keys show-ans, show-pos and save-ref are implemented.

```
4924 \cs_new_protected:Nn \__enumext_starred_item_exec:
4925 {
4926    \tl_put_left:Ne \l__enumext_store_current_label_tl { \l__enumext_label_viii_tl }
4927    \__enumext_store_addto_prop:V \l__enumext_store_current_label_tl
4928    \__enumext_keyans_store_ref:
4929    \tl_put_left:Ne \l__enumext_store_current_label_tl { \item }
602024 by Pablo González L
```

125 / 156

```
\__enumext_keyans_addto_seq_link:
                                        \int_gincr:N \g__enumext_check_starred_cmd_int
                                        \bool_if:NT \l__enumext_show_answer_bool
                                4933
                                               _enumext_print_keyans_box:NN \l__enumext_labelwidth_i_dim \l__enumext_labelsep_i_dim
                                4935
                                        \bool_if:NT \l__enumext_show_position_bool
                                4936
                                4937
                                            \tl_set:Ne \l__enumext_mark_answer_sym_tl
                                4938
                                              {
                                                 \group_begin:
                                                   \exp_not:N \normalfont
                                                   \exp_not:N \footnotesize [ \int_eval:n
                                                       \prop_count:c { g__enumext_ \l__enumext_store_name_tl _prop }
                                4945
                                4946
                                                 \group_end:
                                4947
                                             \__enumext_print_keyans_box:NN \l__enumext_labelwidth_i_dim \l__enumext_labelsep_i_dim
                                      }
                                (End of definition for \__enumext_starred_item_viii:w and others.)
                                The implementation at this is very similar to that of the enumext* environment.
    \__enumext_fake_make_label_viii:n
                                4952 \cs_new_protected_nopar:Npn \__enumext_fake_make_label_viii:n #1
                                        \legacy_if:nT { @noitemarg }
                                4954
                                4955
                                            \legacy_if_set_false:n { @noitemarg }
                                4956
                                            \legacy_if:nT { @nmbrlist }
                                4957
                                4958
                                                 \refstepcounter{enumXviii}
                                4959
                                        \bool_if:NT \l__enumext_item_starred_viii_bool
                                          {
                                            \__enumext_starred_item_exec:
                                          }
                                        \makebox[ \l__enumext_labelwidth_viii_dim ][ \l__enumext_align_label_viii_str ]
                                4966
                                4967
                                            \tl_use:N \l__enumext_label_font_style_viii_tl
                                4968
                                            \bool_if:NTF \l__enumext_wrap_label_viii_bool
                                4969
                                4970
                                                 \__enumext_wrapper_label_viii:n {#1}
                                4971
                                               { #1 }
                                4973
                                        \skip_horizontal:N \l__enumext_labelsep_viii_dim \ignorespaces
                                (End of definition for \__enumext_fake_make_label_viii:n.)
                                13.44.2 Real definition of \item in keyans*
\__enumext_start_item_viii:w
                                The implementation at this is very similar to that of the enumext* environment.
 \__enumext_stop_item_viii:
                                4977 \cs_new_protected_nopar:Npn \__enumext_start_item_viii:w [#1]
                                4978
                                        \cs_set_eq:NN \__enumext_stop_item_tmp_viii: \__enumext_stop_item_viii:
                                4979
                                        \hbox_set_to_wd:Nnw \l__enumext_item_text_viii_box
                                4980
                                          {
                                4981
                                            \l__enumext_joined_width_viii_dim
                                            + \l__enumext_labelwidth_viii_dim
                                            + \l__enumext_labelsep_viii_dim
                                          }
                                          \IfDocumentMetadataTF { }
                                            {
                                               \bool_if:NF \l__enumext_footnotes_key_bool
                                4988
                                                 {
                                                   \__enumext_renew_footnote:
```

```
}
          \__enumext_start_list_tag:n {keyans*}
          \__enumext_fake_make_label_viii:n {#1}
          \__enumext_stop_start_list_tag:
          \__enumext_minipage:w [ t ]{ \l__enumext_joined_width_viii_dim }
4996
            \dim_set_eq:NN \parindent \l__enumext_listparindent_viii_dim
4997
            \skip_set_eq:NN \parskip \l__enumext_parsep_viii_skip
4998
            \__enumext_unskip_unkern:
            \__enumext_unskip_unkern:
            \skip_horizontal:n { -\l__enumext_listparindent_viii_dim } \ignorespaces
            \tl_use:N \l__enumext_fake_item_indent_viii_tl
            \bool_if:NT \l__enumext_item_starred_viii_bool
                 \__enumext_keyans_show_item_opt:
              }
5006
            \tl_use:N \l__enumext_after_list_args_viii_tl
5007
5008
   \cs_new_protected_nopar:Nn \__enumext_stop_item_viii:
5010
          \__enumext_endminipage:
5011
        \__enumext_stop_list_tag:n {keyans*}
       \hbox_set_end:
5013
       \int_set:Nn \hbadness { 10000 }
        \box_use_drop:N \l__enumext_item_text_viii_box
        \IfDocumentMetadataTF { }
5016
          {
5017
            \bool_if:NF \l__enumext_footnotes_key_bool
5018
5019
                \__enumext_print_footnote:
          }
       \int_compare:nNnTF
          { \l_enumext_item_column_pos_viii_int } = { \l_enumext_columns_viii_int }
            \par\noindent
5026
            \int_zero:N \l__enumext_item_column_pos_viii_int
5027
          }
            \skip_horizontal:N \l__enumext_columns_sep_viii_dim
          }
5031
(\textit{End of definition for } \verb|\_-enumext\_start\_item\_viii:w | \textit{and } \verb|\_-enumext\_stop\_item\_viii:.)
```

_enumext_remove_extra_parsep_viii:

The implementation at this is very similar to that of the enumext* environment.

```
5033 \cs_new_protected:Nn \__enumext_remove_extra_parsep_viii:
       \int_compare:nNnT
5035
5036
            \int_mod:nn
5037
              { \g__enumext_item_count_all_viii_int }
5038
              { \l__enumext_columns_viii_int }
         }
         =
5041
         { 0 }
            \para_end:
            \skip_vertical:n { -\l__enumext_itemsep_viii_skip }
            \skip_vertical:N \c_zero_skip
            \int_gzero:N \g__enumext_item_count_all_viii_int
5047
5048
```

 $(\mathit{End}\ of\ definition\ for\ \verb|_-enumext_remove_extra_parsep_viii:.)$

13.45 The command \getkeyans

\getkeyans __enumext_getkeyans_aux:n __enumext_getkeyans:nn

The \getkeyans command takes a mandatory argument of the form $\{\langle store\ name: position \rangle\}$. Retrieve a "single content" stored by \anskey, \anspic* and \item* and anskey* from prop list defined by save-ans key.

```
5050 \NewDocumentCommand \getkeyans { m }
©2024 by Pablo González L
```

The internal function __enumext_getkeyans_aux:n is in charge of *splitting* the *mandatory argument* using ":". If ":" is omitted it will return an error.

```
5055 \cs_new_protected:Npn \__enumext_getkeyans_aux:n #1
5056
     {
       \str_if_in:nnTF {#1} { : }
5057
         {
           \use:e
              {
                \cs set:Npn \exp not:N \ enumext tmp:w ##1 \c colon str ##2 \scan stop:
5061
                  { {##1} {##2} }
5062
5063
            \exp_after:wN \__enumext_getkeyans:nn \__enumext_tmp:w #1 \scan_stop:
5064
         { \msg_error:nnn { enumext } { missing-colon } {#1} }
5066
5067
```

The internal function __enumext_getkeyans:nn will check for the existence of the *prop list*, if it does not exist it will return an error message, then it will fetch the content specified by the *second argument* from *prop list*.

(End of definition for \getkeyans , __enumext_getkeyans_aux:n, and __enumext_getkeyans:nn. This function is documented on page 17.)

13.46 The command \printkeyans

The \printkeyans command prints "all stored content" in the sequence defined by the save-ans key. The first thing we will do is define a set of \(\frac{filtered keys} \) with which we will control the options of the different nesting levels for the environment enumext and enumext* by storing their values in the list of tokens \\l_enumext_print_keyans_X_tl.

The variable \l__enumext_print_keyans_starred_tl will have the default $\langle keys \rangle$ for \printkeyans* and will be set by \setenumext[$\langle print^* \rangle$] and the variable \l__enumext_print_keyans_vii_tl will have the default keys for the environment enumext* nested within the sequence and will be set by \setenumext[$\langle print, * \rangle$], the rest of the variables will be for the environment enumext and will be set by \setenumext[$\langle print, * \rangle$].

```
\keys_define:nn { enumext / print }
    {
                            = \keys_precompile:neN { enumext / enumext* }
       print*
               .code:n
                                { \__enumext_filter_save_key:n {#1} }
5081
                                \l__enumext_print_keyans_starred_tl, % starred cmd
5082
       print* .initial:n
                           = { nosep, label=\arabic*., columns=2, first=\small, font=\small },
5083
       print-1 .code:n
                            = \keys_precompile:neN { enumext / level-1 }
5084
                                { \__enumext_filter_save_key:n {#1} }
5085
                                \l__enumext_print_keyans_i_tl,
5086
       print-1 .initial:n = { nosep, label=\arabic*., columns=2, first=\small, font=\small },
5087
       print-2 .code:n
                            = \keys_precompile:neN { enumext / level-2 }
                                { \__enumext_filter_save_key:n {#1} }
                                \l__enumext_print_keyans_ii_tl,
       print-2 .initial:n = { nosep, label=(\alph*), first=\small, font=\small },
       print-3 .code:n
                            = \keys_precompile:neN { enumext / level-3 }
                                { \__enumext_filter_save_key:n {#1} }
5093
                                \l__enumext_print_keyans_iii_tl,
5094
       print-3 .initial:n = { nosep, label=\roman*., first=\small, font=\small },
5095
                            = \keys_precompile:neN { enumext / level-4 }
       print-4 .code:n
5096
                                { \__enumext_filter_save_key:n {#1} }
5097
                                \l__enumext_print_keyans_iv_tl,
       print-4 .initial:n = { nosep, label=\Alph*., first=\small, font=\small },
                            = \keys_precompile:neN { enumext / enumext* }
       print-* .code:n
```

The reason for storing \(\lambda eys \rangle \) in token lists using \(\lambda eys_precompile: neN \) is because the keys are set via \(\setenumext \) but are later executed by running the command \(\printkeyans \) and they are not handled directly by its optional argument, except those related to the \(first \) opening level.

\printkeyans

__enumext_printkeyans:nnn

Create a user command to print "all stored content" in sequence for \anskey, anskey*, \item* and \anspic*. Within a group we will run our "precompiled keys" and then call the internal function __enumext_-printkeyans:nnn.

```
NewDocumentCommand \printkeyans { s O{} m }

{

group_begin:

\tl_use:N \l_enumext_print_keyans_i_tl

\tl_use:N \l_enumext_print_keyans_ii_tl

\tl_use:N \l_enumext_print_keyans_iii_tl

\tl_use:N \l_enumext_print_keyans_iv_tl

\tl_use:N \l_enumext_print_keyans_vii_tl

\tl_use:N \l_enumext_print_keyans_vii_tl

\_enumext_printkeyans:nnn { #1 } { #2 } { #3 }

\group_end:
```

The internal function __enumext_printkeyans:nnn will check for the existence of the *sequence*, if it does not exist it will return an error message, then it will check if not empty.

If the starred argument '*' is present we will check that the environment enumext* is not saved in the sequence, then execute the variable \l__enumext_print_keyans_starred_tl that contains the default $\langle keys \rangle$ for the environment enumext*, we set \l__enumext_base_line_fix_bool and \l__enumext_print_keyans_star_bool to true for baseline correction, open the enumext* environment passing the optional argument and map the sequence, then set \l__enumext_base_line_fix_bool and \l__enumext_print_keyans_star_bool to false.

Otherwise it will open the environment enumext passing the *optional argument* to the "first level" then map the *sequence*.

 $(\textit{End of definition for } \verb|\printkeyans| and \verb|\printkeyans| in This function is documented on page 18.)$

13.47 The command \setenumext

The command \setenumext will be in charge of managing the $\langle keys \rangle$ passed to all environments and to the \printkeyans command. We must take precautions with the enumext* environment and "first level" of the enumext environment so as not to capture $\langle keys \rangle$ that complicate us.

__enumext_filter_first_level:n __enumext_filter_first_level_key:n __enumext_filter_first_level_pair:nn The function $_$ enumext_filter_first_level:n will be in charge of filtering the $\langle keys \rangle$ passed to the environment enumext* and "first level" of the environment enumext.

The function __enumext_filter_first_level_key:n will be responsible for filtering the $\langle keys \rangle$ that are passed "without value" by excluding the keys resume and resume*.

The function $\ensuremath{\mbox{\mbox{$\setminus$}}}$ enumext_filter_first_level_pair:nn will be responsible for filtering the $\langle \textit{keys} \rangle$ that are passed "with value" by excluding the series, resume and save-ans keys.

Now define a "meta families" of $\langle keys \rangle$ to access from \setenumext.

```
\keys_define:nn { enumext / meta-families }
       enumext-1 .code:n =
                      \keys_set:ne { enumext / level-1 }
5182
                             _enumext_filter_first_level:n {#1}
5185
                    } ,
5186
       enumext-2 .code:n = { \keys_set:nn { enumext / level-2 } {#1} } ,
5187
       enumext-3 .code:n = { \keys_set:nn { enumext / level-3 } {#1} } ,
5188
       enumext-4 .code:n = { \keys_set:nn { enumext / level-4 } {#1} } ,
       kevans
                  .code:n = { \keys_set:nn { enumext / keyans } {#1} } ,
       enumext*
                  .code:n =
                    {
                      \keys_set:ne { enumext / enumext* }
                            _enumext_filter_first_level:n {#1}
5196
                   },
       kevans*
                  .code:n = { \keys_set:nn { enumext / keyans* } {#1} } ,
5198
                  .code:n = { \keys_set:nn { enumext / print
                                                                 } { print*
       print*
5199
       print-1
                  .code:n = { \keys_set:nn { enumext / print
                                                                 } { print-1 = {#1} } } ,
       print-2
                  .code:n = { \keys_set:nn { enumext / print
                                                                 } { print-2 = {#1} } } ,
                  .code:n = { \keys_set:nn { enumext / print
       print-3
                                                                 } { print-3 = {#1} } } ,
                  .code:n = { \keys_set:nn { enumext / print
                                                                } { print-4 = {#1} } } ,
       print-4
©2024 by Pablo González L
```

130 / 156

```
print-*
                                                                                                                                                                        .code:n = { \keys_set:nn { enumext / print } { print-* = {#1} } } ,
                                                                                                                                                                       .code:n = { \msg_error:nn { enumext } { unknown-key-family } } ,
                                                                                                                             unknown
                                                                                                  We store them in the constant sequence \c__enumext_all_families_seq separated by commas.
                                                                                                   5207 \seq const from clist:Nn \c enumext all families seq
                                                                                                                              enumext-1, enumext-2, enumext-3, enumext-4, keyans, enumext*,
                                                                                                                              keyans*, print-1, print-2, print-3, print-4, print-*, print*,
                                                                                                  5211
                                                                                                Now we define the user command \setenumext.
                                              \setenumext
   \__enumext_set_parse:n
                                                                                                  \NewDocumentCommand \setenumext { O{enumext,1} +m }
\__enumext_set_error:nn
                                                                                                                               \seq_clear:N \l__enumext_setkey_tmpa_seq
                                                                                                                               \seq_set_from_clist:Nn \l__enumext_setkey_tmpb_seq {#1}
                                                                                                                               \int_set:Nn \l__enumext_setkey_tmpa_int
                                                                                                  5216
                                                                                                                                               \seq_count:N \l__enumext_setkey_tmpb_seq
                                                                                                                                      }
                                                                                                  5219
                                                                                                                               \int_compare:nNnTF { \l__enumext_setkey_tmpa_int } > { 1 }
                                                                                                                                               \seq_pop_left:NN \l__enumext_setkey_tmpb_seq \l__enumext_setkey_tmpa_tl
                                                                                                                                               \seq_set_map_e:NNn \l__enumext_setkey_tmpa_seq \l__enumext_setkey_tmpa_seq
                                                                                                                                                      {
                                                                                                                                                              \tl use:N \l enumext setkey tmpa tl - ##1
                                                                                                  5227
                                                                                                                                      }
                                                                                                  5228
                                                                                                                                       {
                                                                                                                                               \seq_put_right:Ne \l__enumext_setkey_tmpa_seq { \tl_trim_spaces:n {#1} }
                                                                                                                               \seq_if_empty:NTF \l__enumext_setkey_tmpa_seq
                                                                                                                                      { \seq_map_inline:Nn \c__enumext_all_families_seq }
                                                                                                                                       { \seq_map_inline:Nn \l__enumext_setkey_tmpa_seq }
                                                                                                                                               \keys_set:nn { enumext / meta-families } { ##1 = {#2} }
                                                                                                   5236
                                                                                                                                      }
                                                                                                  5237
                                                                                                  5238
                                                                                                  Internal functions used by the \setenumext command.
                                                                                                   5239 \cs_new_protected:Npn \__enumext_set_parse:n #1
                                                                                                  5240
                                                                                                                               \tl_set:Ne \l__enumext_setkey_tmpb_tl { \tl_trim_spaces:n {#1} }
                                                                                                  5241
                                                                                                                              \clist_map_inline:nn { 0, 1, 2, 3, 4, * } % <- max level
                                                                                                  5242
                                                                                                                                      { \tl_remove_all:Nn \l__enumext_setkey_tmpb_tl {##1} }
                                                                                                  5243
                                                                                                                               \tl_if_empty:NTF \l__enumext_setkey_tmpb_tl
                                                                                                                                               \seq_put_right:Ne \l__enumext_setkey_tmpa_seq
                                                                                                                                                       { \tl_trim_spaces:n {#1} }
                                                                                                                                      }
                                                                                                                                       { \ \ } { \ \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { \ \ } { 
                                                                                                   5249
                                                                                                  5250
                                                                                                              \cs_new_protected:Npn \__enumext_set_error:nn #1 #2
                                                                                                                       { \msg_error:nnn { enumext } { invalid-key } {#1} {#2} }
                                                                                                 (\textit{End of definition for } \texttt{\ } \texttt
                                                                                                  page 6.)
```

13.48 The command \setenumextmeta

The command \setenumextmeta will be responsible for adding new "meta-keys" for the enumext and enumext* environments. The implementation code was given by Jonathan P. Spratte (@Skillmon) answer in Add .meta key to existing keys (l3keys).

```
\setenumextmeta
```

First we will create a prop list \c_enumext_meta_paths_prop to handle the optional argument.

```
{enumext*} = enumext*
```

Now we create the user command taking care that unknown cannot be passed as an argument.

```
NewDocumentCommand \setenumextmeta { s O{enumext,1} m +m }
5262
       \str_if_eq:eeTF { \tl_trim_spaces:n {#3} } { unknown }
5263
         { \msg_error:nn { enumext } { prohibited-unknown } }
5265
           \bool_if:nTF {#1}
5266
             {
5267
               \int_step_inline:nn { 4 }
5268
                 { \__enumext_add_meta_key:nnn { enumext, ##1 } {#3} {#4} }
               \__enumext_add_meta_key:nnn { enumext* } {#3} {#4}
             { \__enumext_add_meta_key:nnn {#2} {#3} {#4} }
         }
```

The internal functions __enumext_add_meta_key:nnn and __enumext_def_meta_key:nnn will check the *optional argument* and create the *"meta-key"*.

```
\cs_new_protected:Npn \__enumext_add_meta_key:nnn #1
5276
      \tl_set:Nn \l__enumext_meta_path_tl {#1}
      \tl_replace_all:Nnn \l__enumext_meta_path_tl { ~ } {}
      \prop_get:NVNTF
        { \__enumext_def_meta_key:Vnn \l__enumext_meta_path_tl }
5281
5282
        {
           \msg_error:nnn { enumext } { unknown-set } {#1}
5283
           \use none:nn
5284
5285
5286
   \cs_new_protected:Npn \__enumext_def_meta_key:nnn #1#2#3
5287
5288
      \bool_lazy_or:nnTF
5289
        { \keys_if_exist_p:nn { enumext / #1 } {#2} }
        { \keys_if_exist_p:nn { enumext / enumext* } {#2} }
5291
        { \msg_error:nnn { enumext } { already-defined } {#2} }
5292
5293
          \keys_define:nn { enumext / #1 }
5294
            {
5295
              #2 .meta:n = \{ #3 \},
5296
              #2 .value_forbidden:n = true
        }
_{5301} \cs_generate_variant:Nn \__enumext_def_meta_key:nnn { V }
```

(End of definition for \setenumextmeta and others. This function is documented on page 6.)

13.49 The command \foreachkeyans

The command \foreachkeyans will execute a *loop* over the *prop list* and return its contents. The implementation code is adapted from the answer provided by Enrico Gregorio (@egreg) in Expand a .cs defined by key inside the function.

\foreachkeyans

```
\__enumext_parse_foreach_keys:nn
\__enumext_parse_foreach_keys:n
\__enumext_foreach_keyans:nn
\__enumext_foreach_add_body:n
```

We define a set of $\langle keys \rangle$ for command and we will save the default values of these in $\g_enumext_-$ foreach_default_keys_tl to avoid the use of group.

132 / 156

```
5302 \keys_define:nn { enumext / foreach }
       before .tl_set:N = \l__enumext_foreach_before_tl,
       before .value_required:n = true,
       after
               .tl_set:N = \l__enumext_foreach_after_tl,
5306
       after
               .value_required:n = true,
5307
       start
               .int_set:N = \l__enumext_foreach_start_int,
5308
       start
               .value_required:n = true,
5309
               .int_set:N = \l__enumext_foreach_stop_int,
       stop
                .value_required:n = true,
       stop
               .int_set:N = \l__enumext_foreach_step_int,
       step
                .value_required:n = true,
       wrapper .cs_set_protected:Np = \__enumext_foreach_wrapper:n #1,
       wrapper .value_required:n = true,
©2024 by Pablo González L
```

.tl_set:N = \l__enumext_foreach_sep_tl,

sep

sep

```
.value_required:n = true,
                          = { \__enumext_parse_foreach_keys:n {#1} }
       unknown .code:n
   \keys_precompile:nnN { enumext / foreach }
       before={},after={},start=1,step=1,stop=0,wrapper=#1,sep={; }
5322
     \g__enumext_foreach_default_keys_tl
Functions for handling unknown \langle keys \rangle.
   \cs_new_protected:Npn \__enumext_parse_foreach_keys:nn #1#2
5326
       \tl_if_blank:nTF {#2}
5327
5328
            \msg_error:nnn { enumext } { for-key-unknown } {#1}
            \msg_error:nnnn { enumext } { for-key-value-unknown } {#1} {#2}
\cs_new_protected:Npn \__enumext_parse_foreach_keys:n #1
5336
       \exp_args:NV \__enumext_parse_foreach_keys:nn \l_keys_key_str {#1}
5337
5338
We create the command.
5339 \NewDocumentCommand \foreachkeyans { +O{} m }
       \__enumext_foreach_keyans:nn {#1} {#2}
5341
5342
Finally the internal functions \__enumext_foreach_keyans:nn and \__enumext_foreach_add_body:n
will loop through the prop list and print the contents.
<sub>5343</sub> \cs_new_protected:Npn \__enumext_foreach_keyans:nn #1 #2
       \tl_use:N \g__enumext_foreach_default_keys_tl
5345
       \keys_set:nn { enumext / foreach } {#1}
5346
       \tl_set:Nn \l__enumext_foreach_name_prop_tl {#2}
5347
       \prop_if_exist:cF { g__enumext_#2_prop }
5348
            \msg_error:nnn { enumext } { undefined-storage-anskey } {#2}
       \int_compare:nNnT { \l__enumext_foreach_stop_int } = { 0 }
            \int set:Nn \l enumext foreach stop int
              { \prop_count:c { g__enumext_#2_prop } }
       \seq_clear:N \l__enumext_foreach_print_seq
       \int_step_function:nnnN
          { \l__enumext_foreach_start_int }
          { \l__enumext_foreach_step_int }
          { \l__enumext_foreach_stop_int }
         \__enumext_foreach_add_body:n
         \verb|\seq_use:NV \ | l_enumext_for each\_print_seq \ | l_enumext_for each\_sep\_tl| \\
5364
   \cs_new_protected:Npn \__enumext_foreach_add_body:n #1
5365
5366
       \seq_put_right:Ne \l__enumext_foreach_print_seq
5367
5368
            \exp_not:V \l__enumext_foreach_before_tl
5369
            \__enumext_foreach_wrapper:n
5371
                \prop_item:cn { g__enumext_ \l__enumext_foreach_name_prop_tl _prop }{#1}
            \exp_not:V \l__enumext_foreach_after_tl
     }
5376
(End of definition for \foreachkeyans and others. This function is documented on page 17.)
```

13.50 Messages

```
Message used by package-load for multicol and hyperref packages.
```

```
5377 \msg_new:nnn { enumext } { package-load }
       The ~ '#1' ~ package ~ is ~ already ~ loaded.
5379
5380
5381 \msg_new:nnn { enumext } { package-not-load }
5382
       The ~ '#1' ~ package ~ will ~ be ~ loaded ~ as ~ a ~ dependency.
s385 \msg_new:nnn { enumext } { package-load-foot }
       The ~ '#1' ~ package ~ is ~ loaded ~ with ~ the ~ option ~ '#2'.
5387
5388
Message used in the creation of counters by enumext package.
<sub>5389</sub> \msg_new:nnn { enumext } { counters }
       The ~ counter ~ '#1' ~ is ~ already ~ defined ~ by ~ some ~ \\
5391
       package ~ or ~ macro, ~ it ~ cannot ~ be ~ continued.
5392
5393
Message used by align and mark-pos keys.
5394 \msg_new:nnn { enumext } { unknown-choice }
       The ~ value ~ '#3' ~ for ~ '#1' ~ key ~ is ~ invalid ~ use ~ ('#2').
5396
5397
Message used by reserved anskey* environment by enumext package.
5398 \msg_new:nnnn { enumext } { anskey-env-error }
5399
       The ~ '#1' ~ environment ~is ~ reserved ~ by ~\\
       'enumext' ~ package, ~ It~ is~ already~ defined.
       The ~ anskey* ~ environment ~ is ~ defined ~ internally ~
       for ~ the ~ 'save-ans' ~ key.\\
Message used in the creation of prop list by enumext package.
5407 \msg_new:nnn { enumext } { store-prop }
       * ~ Package ~ enumext: ~ Creating ~
        \c_backslash_str g__enumext_#1_prop ~ \msg_line_context:.
5410
     }
_{\text{5412}} \ \text{\sc msg\_new:nnn} \ \{ \ \text{enumext} \ \} \ \{ \ \text{store-seq} \ \}
5413
        * ~ Package ~ enumext: ~ Creating ~
5414
       \c_backslash_str g__enumext_#1_seq ~ \msg_line_context:.
5415
     }
5416
5417 \msg_new:nnn { enumext } { store-int }
5418
        * ~ Package ~ enumext: ~ Creating ~
5419
       \c_backslash_str g__enumext_resume_#1_int ~ \msg_line_context:.
5420
5422 \msg_new:nnn { enumext } { prop-seq-int-hook }
        * ~ Package ~ enumext: ~ Elements ~ in ~
       \c_backslash_str g__enumext_#1_prop ~ = ~ #2.\\
       * ~ Package ~ enumext: ~ Elements ~ in ~
5426
       \c_backslash_str g__enumext_#1_seq ~ = ~ #3.\\
       * ~ Package ~ enumext: ~ Value ~ off ~
5428
       \c_backslash_str g__enumext_resume_#1_int ~ = ~ #4.
5429
5430
5431 \msg_new:nnn { enumext } { item-answer-hook }
        * ~ Package ~ enumext: ~ Value ~ off ~
       \c_backslash_str g__enumext_item_number_int ~ = ~ #1.\\
       * ~ Package ~ enumext: ~ Value ~ off ~
       \c_backslash_str g__enumext_item_anskey_int ~ = ~ #2.\\
        * ~ Package ~ enumext: ~ Difference ~ item_number_int ~ - ~ item_anskey_int ~ = ~ #3.
5437
```

```
Message used by [\langle key = val \rangle] system and \setenumext command.
5439 \msg_new:nnn { enumext } { invalid-key }
       The \sim key \sim '#1' \sim is \sim not \sim know \sim the \sim level \sim #2.
5441
5442
5443 \msg_new:nnn { enumext } { unknown-key-family }
5444
       Unknown~key~family~`\l_keys_key_str'~for~enumext.
5445
Messages used in length calculation.
5447 \msg_new:nnn { enumext } { width-negative }
5448
       Ignoring ~ negative ~ value ~ '#1=#2' ~ \msg_line_context:.\\
5449
       The ~ key ~ '#1'~ accepts ~ values ~ >= ~ Opt.
5450
5451
   \msg_new:nnn { enumext } { width-zero }
5453
       Invalid ~ '#1=#2' ~ \msg_line_context:.\\
       The ~ key ~ '#1'~ accepts ~ values ~ > ~ Opt.
5455
5456
Messages used by show-length key in enumext.
5457 \msg_new:nnn { enumext } { list-lengths }
       **** ~ Lengths ~ used ~ by ~ 'enumext' ~ level ~ '#2' ~ \msg_line_context:~\c_space_tl ****\\
       \__enumext_show_length:nnn { dim } { labelsep
                                                            } {#1}
       \__enumext_show_length:nnn { dim } { labelwidth } {#1}
       \__enumext_show_length:nnn { dim } { itemindent
                                                              } {#1}
       \__enumext_show_length:nnn { dim } { leftmargin } {#1}
5463
       \__enumext_show_length:nnn { dim } { rightmargin } {#1}
       \__enumext_show_length:nnn { dim } { listparindent } {#1}
       \__enumext_show_length:nnn { skip } { topsep
       \__enumext_show_length:nnn { skip } { parsep
       \__enumext_show_length:nnn { skip } { partopsep } {#1}
       \label{lem:length:nnn} $$\sum_{\substack{n=1\\ n \in \mathbb{N}}} { \text{ itemsep } } { \text{ $\#1$} }
5470
5471
Messages used by show-length key in enumext*, keyans* and keyans.
5472 \msg_new:nnn { enumext } { list-lengths-not-nested }
       **** ~ Lengths ~ used ~ by ~ '#2' ~ environment ~ \msg_line_context:~\c_space_tl ****\\
5474
       \__enumext_show_length:nnn { dim } { labelsep
                                                            } {#1}
5475
       \__enumext_show_length:nnn { dim } { labelwidth
                                                              } {#1}
5476
       \__enumext_show_length:nnn { dim } { itemindent
                                                              } {#1}
5477
       \__enumext_show_length:nnn { dim } { leftmargin
                                                              } {#1}
5478
       \__enumext_show_length:nnn { dim } { rightmargin
5479
       \__enumext_show_length:nnn { dim } { listparindent } {#1}
       \__enumext_show_length:nnn { skip } { topsep
       \__enumext_show_length:nnn { skip } { parsep
       \__enumext_show_length:nnn { skip } { partopsep } {#1}
       \__enumext_show_length:nnn { skip } { itemsep } {#1}
5484
Messages used by ref key.
5487 \msg_new:nnn { enumext } { key-ref-empty }
       Key ~ 'ref' ~ need ~ a ~ value ~ in ~ '#1'~ \msg_line_context:.
Messages used by save-ans key.
save-ans-empty }
       Key ~ 'save-ans' ~ need ~ a ~ value ~ in ~ '#1'~ \msg_line_context:.
5495 \msg_new:nnn { enumext } { save-ans-log }
        ~ Package ~ enumext: ~ Start ~ #1\c_space_tl with ~ save-ans=#2 ~ \msg_line_context:.
5499 \msg_new:nnn { enumext } { save-ans-log-hook }
©2024 by Pablo González L
```

```
* ~ Package ~ enumext: ~ Stop ~ #1\c_space_tl with ~ save-ans=#2 ~ \msg_line_context:.
\msg_new:nnn { enumext } { save-ans-hook }
5504
       Stop ~ storing ~ for ~ 'save-ans=#1' ~ \msg_line_context:.
5505
     }
5506
Messages used by the internal system to check answer used by check-ans key.
5507 \msg_new:nnn { enumext } { need-save-ans }
5508
       Key ~ '#1'~ works ~ only ~ with ~ the ~ 'save-ans' ~ key ~ in ~ '#2'~ \msg_line_context:.
5509
5510
5511 \msg_new:nnn { enumext } { items-same-answer }
5512
       ***********
5513
       * ~ Package ~ enumext: ~ Checking ~ answers ~ in ~ '#1' ~
5514
       for ~ \c_left_brace_str #2 \c_right_brace_str\\
5515
       * ~ started ~ #3 ~ and ~ close ~ \msg_line_context: : ~
       'OK', ~ all ~ items ~ with ~ answer.\\
5517
5518
     }
5519
5520 \msg_new:nnn { enumext } { item-greater-answer }
5521
       Checking ~ answers ~ in ~ '#1' ~ for ~ \c_left_brace_str #2 \c_right_brace_str\\
       started ~ #3 ~ and ~ close ~ \msg_line_context: : ~'NOT ~ OK'\\
       Items ~ > ~ Answers.
5525
_{5526} \msg_new:nnn { enumext } { item-less-answer }
       Checking ~ answers ~ in ~ '#1' ~ for ~ \c_left_brace_str #2 \c_right_brace_str\\
5528
       started ~ #3 ~ and ~ close ~ \msg_line_context: : ~'NOT ~ OK'\\
       Items ~ < ~ Answers.</pre>
5530
     }
5531
Messages used by the internal system to check for "starred" \item* and \anspic* commands.
5532 \msg_new:nnn { enumext } { missing-starred }
5533
       Missing ~ '\c_backslash_str #1*' ~ #2.
5534
5535
5536 \msg_new:nnn { enumext } { many-starred }
5537
       Many ~ '\c_backslash_str #1*' ~ #2.
5538
Messages used by \printkeyans* command.
5540 \msg_new:nnn { enumext } { print-starred }
5541
       \c_backslash_str printkeyans*:~ The ~ sequence ~ '#1' ~ already ~ contains ~
5542
       #2 ~ environment ~ \msg_line_context:.
5543
Message for the nesting depth of the environment enumext.
 \msg_new:nnn { enumext } { list-too-deep }
5546
       Too ~ deep ~ nesting ~ for ~ 'enumext' ~ \msg_line_context:.~ \\
5547
       The ~ maximum ~ level ~ of ~ nesting ~ is ~ 4.
Messages used by \anskey, anskey* and \anspic commands.
\msg_new:nnn { enumext } { anskey-unnumber-item }
       Can't ~ store ~ with ~ a ~ unnumbered ~ \c_backslash_str item ~ \msg_line_context:.
_{\text{5554}} \ \mbox{\sc msg\_new:nnn} \ \{ \ \mbox{\sc enumext} \ \} \ \{ \ \mbox{\sc anskey-already-stored} \ \}
       Content ~ already ~ stored ~ for ~ this ~ \c_backslash_str item ~ \msg_line_context:.
5556
5557
5558 \msg_new:nnn { enumext } { anskey-empty-arg }
5559
       Can't ~ store ~ empty ~ content ~ \msg_line_context:.
5560
5561
5562 \msg_new:nnn { enumext } { anskey-wrong-place }
```

```
Wrong ~ place ~ for ~ command ~ '\c_backslash_str #1' ~ \msg_line_context:.~ \\
        '\c_backslash_str #1' ~ works ~ in ~ the ~ environment ~ '#2'.
s567 \msg_new:nnn { enumext } { anskey-nested }
       The ~ command ~ \c_backslash_str anskey~ can't ~ be ~ nested ~ \msg_line_context:.
5569
   \msg_new:nnn { enumext } { anskey-math-mode }
       #1 ~ can't ~ work ~ in ~ math ~ mode ~ \msg_line_context:.
5573
5575 \msg_new:nnn { enumext } { anskey-env-wrong }
       The ~ environment ~ anskey* ~ cannot ~ use ~ in ~ '#1' ~ \msg_line_context:.
5578
s579 \msg_new:nnn { enumext } { anspic-wrong-place }
5580
       Wrong ~ place ~ for ~ command ~ '\c_backslash_str #1' ~ \msg_line_context:.~ \\
5581
         \c_backslash_str #1' ~ works ~ in ~ the ~ environment ~ '#2'.
5582
5583
   \msg_new:nnn { enumext } { command-wrong-place }
5585
       Wrong ~ place ~ for ~ command ~ '\c_backslash_str #1' ~ \msg_line_context:.~ \\
        '\c_backslash_str #1' ~ works ~ outside ~ the ~ environment ~ '#2'.
5587
5588
5589 \msg_new:nnnn { enumext } { anskey-env-key-unknown }
5590
       The ~ key ~ '#1' ~ is ~ unknown ~ by ~ environment~
5591
       'anskey*' ~ and ~ is ~ being ~ ignored.
5592
5593
5594
       The ~ environment ~ 'anskey*' ~ does ~ not ~ have ~ a ~ key ~ called ~'#1'.\\
       Check ~ that ~ you ~ have ~ spelled ~ the ~ key ~ name ~ correctly.
5598 \msg_new:nnnn { enumext } { anskey-env-key-value-unknown }
       The \sim key \sim '#1=#2' \sim is \sim unknown \sim by \sim environment \sim
5600
       'anskey*' ~ and ~ is ~ being ~ ignored.
5601
5602
       The ~ environment ~ 'anskey*' ~ does ~ not ~ have ~ a ~ key ~ called ~'#1'.\\
       Check \sim that \sim you \sim have \sim spelled \sim the \sim key \sim name \sim correctly.
   \msg_new:nnnn { enumext } { anskey-cmd-key-unknown }
     { The \sim key \sim'\#1'\sim is \sim unknown \sim by \sim '\c_backslash_str anskey' \sim and \sim is \sim being \sim ignored.}
       The ~ command ~'\c_backslash_str anskey' ~ does ~ not ~ have ~ a ~ key ~ called ~'#1'.\\
       Check ~ that ~ you ~ have ~ spelled ~ the ~ key ~ name ~ correctly.
   \msg_new:nnnn { enumext } { anskey-cmd-key-value-unknown }
     { The ~ key ~ '#1=#2' ~ is ~ unknown ~ by ~ '\c_backslash_str anskey' ~ and ~ is ~ being ~ igno
       The ~ command ~ '\c_backslash_str anskey' ~ does ~ not ~ have ~ a ~ key ~ called ~'#1'.\\
       Check ^{\sim} that ^{\sim} you ^{\sim} have ^{\sim} spelled ^{\sim} the ^{\sim} key ^{\sim} name ^{\sim} correctly.
Messages used by keyans, keyans* and keyanspic environment.
5619 \msg_new:nnn { enumext } { keyans-nested }
       The ~ environment ~ 'keyans' ~ can't ~ be ~ nested ~ \msg_line_context:.
5623 \msg_new:nnn { enumext } { keyans-wrong-level }
5624
       Wrong ~ level ~ position ~ for ~ 'keyans' ~ \msg_line_context:.~ \\
5625
       The ~ environment ~ 'keyans' ~ can ~ only ~ be ~ in ~ the ~ first ~ level.
5626
5627
5628 \msg_new:nnn { enumext } { wrong-place }
5629
       Wrong ~ place ~ for ~ '#1' ~ environment ~\msg_line_context:.~ \\
5630
       '#1' ~ is ~ only ~ found ~ with ~ '#2' ~ in ~ 'enumext.
5633 \msg_new:nnn { enumext } { keyanspic-nested }
```

```
The ~ environment ~ 'keyanspic' ~ can't ~ be ~ nested~ \msg_line_context:.~.
5635
5636
5637 \msg_new:nnn { enumext } { keyanspic-wrong-level }
       Wrong ~ level ~ position ~ for ~ 'keyanspic' ~ \msg_line_context:.~ \\
       The ~ environment ~ 'keyans' ~ can ~ only ~ be ~ in ~ the ~ first ~ level.
   \msg_new:nnn { enumext } { keyanspic-item-cmd }
       Can't ~ use ~ \c_backslash_str item ~ in ~ keyanspic ~ \msg_line_context:.
   \msg_new:nnnn { enumext } { keyans-unknown-key }
5647
       The ~ key ~ '#1' ~ is ~ unknown ~ by ~ environment~
5648
        '\l__enumext_envir_name_tl' ~ and ~ is ~ being ~ ignored.
5649
5650
5651
       The ~ environment ~ '\l__enumext_envir_name_tl' ~ does ~ not
5652
        ~ have ~ a ~ key ~ called ~'#1'.\\
5653
       Check ^{\sim} that ^{\sim} you ^{\sim} have ^{\sim} spelled ^{\sim} the ^{\sim} key ^{\sim} name ^{\sim} correctly.
5655
   \msg_new:nnnn { enumext } { keyans-unknown-key-value }
5657
       The ~ key ~ '#1=#2' ~ is ~ unknown ~ by ~ environment ~
5658
        '\l__enumext_envir_name_tl' ~ and ~ is ~ being ~ ignored.
5659
5660
5661
       The ~ environment ~ '\l__enumext_envir_name_tl' ~ does ~ not
        ~ have ~ a ~ key ~ called ~'#1'.\\
       Check ~ that ~ you ~ have ~ spelled ~ the ~ key ~ name ~ correctly.
5665
Message used by unknown \langle keys \rangle in enumext*. environment.
   \msg_new:nnnn { enumext } { starred-unknown-key }
       The \sim key \sim '#1' \sim is \sim unknown \sim by \sim environment\sim
        '\l__enumext_envir_name_tl' ~ and ~ is ~ being ~ ignored.
       The ~ environment ~ '\l__enumext_envir_name_tl' ~ does ~ not
5672
        ~ have ~ a ~ key ~ called ~'#1'.\\
5673
       Check ~ that ~ you ~ have ~ spelled ~ the ~ key ~ name ~ correctly.
5674
5675
   \msg_new:nnnn { enumext } { starred-unknown-key-value }
5676
5677
       The \sim key \sim '#1=#2' \sim is \sim unknown \sim by \sim environment \sim
5678
        '\l__enumext_envir_name_tl' ~ and ~ is ~ being ~ ignored.
5679
       The ~ environment ~ '\l__enumext_envir_name_tl' ~ does ~ not
       ~ have ~ a ~ key ~ called ~'#1'.\\
5683
       Check ~ that ~ you ~ have ~ spelled ~ the ~ key ~ name ~ correctly.
5684
Message used by unknown \langle \textit{keys} \rangle in enumext environment.
5686 \msg_new:nnnn { enumext } { standar-unknown-key }
       The ~ key ~ '#1' ~ is ~ unknown ~ by ~ environment ~ '\l_enumext_envir_name_tl' \c_space_tl
        ~ on ~ level ~ \ int_use:N \ l_enumext_level_int \ c_space_tl and ~ is ~ being ~ ignored.
       The ~ environment ~ '\l__enumext_envir_name_tl' ~ does ~ not
        ~ have ~ a ~ key ~ called ~'#1' ~ on ~ level ~ \int_use:N \l__enumext_level_int.\\
       Check ^{\sim} that ^{\sim} you ^{\sim} have ^{\sim} spelled ^{\sim} the ^{\sim} key ^{\sim} name ^{\sim} correctly.
5696 \msg_new:nnnn { enumext } { standar-unknown-key-value }
5697
       The ~ key ~ '#1=#2' ~ is ~ unknown ~ by ~ environment ~ '\l_enumext_envir_name_tl' \c_space_
5698
        ~ on ~ level ~ \int_use:N \l__enumext_level_int \c_space_tl and ~ is ~ being ~ ignored.
5699
5700
```

```
The ~ environment ~ '\l__enumext_envir_name_tl' ~ does ~ not
       ~ have ~ a ~ key ~ called ~'#1' ~ on ~ level ~ \int_use:N \l__enumext_level_int.\\
       Check ~ that ~ you ~ have ~ spelled ~ the ~ key ~ name ~ correctly.
5705
Message used by unknown \langle keys \rangle in \foreachkeyans.
5706 \msg_new:nnnn { enumext } { for-key-unknown }
     { The~key~'#1'~is~unknown~by~'\c_backslash_str foreachkeyans'~and~is~being~ignored.}
5708
       The~command~'\c_backslash_str foreachkeyans'~does~not~have~a~key~called~'#1'.\\
5709
       Check~that~you~have~spelled~the~key~name~correctly.
5711
   \msg_new:nnnn { enumext } { for-key-value-unknown }
     { The~key~'#1=#2'~is~unknown~by~'\c_backslash_str foreachkeyans'~and~is~being~ignored. }
       The~command~'\c_backslash_str foreachkeyans'~does~not~have~a~key~called~'#1'.\\
       Check~that~you~have~spelled~the~key~name~correctly.
5716
Messages used by \getkeyans command.
5718 \msg_new:nnn { enumext } { undefined-storage-anskey }
       Storage ~ named ~ '#1' ~ is ~ not ~ defined ~ \msg_line_context:.
Messages used by \miniright command.
5722 \msg_new:nnn { enumext } { missing-miniright }
       Missing ~ '\c_backslash_str miniright' ~ in ~ \msg_line_context:.\\
       The ~ key ~ 'mini-env' ~ need ~ '\c_backslash_str miniright'.
5727 \msg_new:nnn { enumext } { wrong-miniright-place }
5728
       Wrong ~ place ~ for ~ '\c_backslash_str miniright' ~ \msg_line_context:.~ \\
5729
       Works ~ in ~ 'enumext' ~ and ~ 'keyans' ~ with ~ key ~ 'mini-env'.
5730
   \msg_new:nnn { enumext } { wrong-miniright-use }
5732
       Wrong ~ use ~ for ~ '\c_backslash_str miniright' ~ \msg_line_context:.~ \\
       '\c_backslash_str miniright' ~ need ~ a ~ key ~ 'mini-env'.
   \msg_new:nnn { enumext } { wrong-miniright-starred }
5737
5738
       Can't ~ use ~ \c_backslash_str miniright ~ in ~ starred ~ environments ~ \msg_line_context:.
5739
   \msg_new:nnn { enumext } { many-miniright-used }
5742
       Can't ~ use ~ \c_backslash_str miniright ~ more ~ than ~ once ~ \msg_line_context:.
Messages used by \setenumextmeta command.
5745 \msg_new:nnn { enumext } { unknown-set }
5746
       Argument ~ [#1] ~ is ~ unknown ~ by ~ \c_backslash_str setenumextmeta ~ \msg_line_context:.
5747
5748
5749 \msg_new:nnn { enumext } { already-defined }
       The ~ key ~ '#1' ~ is ~ already ~ defined ~ \msg_line_context:.
5753 \msg_new:nnn { enumext } { prohibited-unknown }
5754
       The ~ name ~ 'unknown' ~ can't ~ be ~ chosen~ for ~ a ~ meta ~ key ~ \msg_line_context:.
5756
Messages used by enumext* and keyans* environments.
5757 \msg_new:nnn { enumext } { nested }
5758
       The ~ environment ~ \l_enumext_envir_name_tl \c_space_tl can't ~ be ~ nested ~ \msg_line_con
5760
5761 \msg_new:nnn { enumext } { nested-horizontal }
5762
       The \sim environment \sim \l__enumext_envir_name_tl \c_space_tl can't \sim be \sim nested \sim in \sim '#1' \sim
5763
```

13.51 Finish package

Finish package implementation.

```
_{\rm 5773} \file_input_stop: _{\rm 5774} \langle/package\rangle
```

14 Index of Implementation

The italic numbers denote the pages where the corresponding entry is described, the numbers underlined and all others indicate the line on which they are implemented in the package code.

61.1	
Symbols	2416, 2428, 2443, 2444, 2478, 2519, 2520, 2821, 3011,
* 228	3012, 3049, 3050, 3532, 3534, 3545, 4100, 4101
\+ 220	\bool_lazy_all:nTF 282, 297, 863, 2042, 2068, 2404,
\ 220	2413, 2426, 2441, 3530, 3543
\\ 236, 2779, 4146, 4149, 5391, 5400, 5405, 5425, 5427, 5434,	\bool_lazy_and:nnTF 261, 271, 873, 1520, 1909, 1918,
5436, 5449, 5454, 5459, 5474, 5513, 5515, 5517, 5522,	2082, 2088, 2477, 2484, 2518, 2662, 2674, 2820, 2826,
5523, 5528, 5529, 5547, 5564, 5581, 5586, 5595, 5604,	3010
5610, 5616, 5625, 5630, 5639, 5653, 5663, 5673, 5683,	\bool_lazy_or:nnTF 1971, 1978, 3048, 4099, 5289 \bool_new:N 34, 35, 36, 37, 38, 39, 40, 41, 64, 73, 97, 102,
5693, 5703, 5709, 5715, 5724, 5729, 5734	
	103, 108, 109, 112, 131, 138, 139, 146, 153, 154, 159,
A	161, 162, 176, 188, 190 \bool_not_p:n 262, 272, 867, 2415, 2479, 2485, 2822,
above	2827, 3533, 3546
above*	\bool_set_eq:NN 3122, 3313, 4629, 4885
\addvspace 1163, 1191, 1234, 1237, 1405, 1408, 1505, 1511,	\bool_set_false:N 436, 885, 2016, 2017, 2049, 2054,
1546, 1552, 1573, 1579, 3604, 3765, 3783, 4037, 4041,	2058, 2062, 2075, 2762, 3507, 3652, 3701, 3788, 3926,
4394, 4409, 4455, 4469	3970, 4547, 4573, 4626, 4827, 4882, 5135, 5136
after 993	\bool_set_true: N . 289, 290, 304, 305, 416, 420, 539,
align	900, 1599, 1604, 1866, 1988, 1989, 2261, 2269, 2763,
\Alph	3116, 3118, 3150, 3152, 3309, 3321, 3468, 3506, 3539,
\Alph	3552, 3625, 3698, 3725, 3923, 4362, 4427, 4546, 4633,
\alph 38, 43, 44	4640, 4641, 4685, 4825, 4889, 4896, 4897, 5130, 5131
\alph 489, 614, 5091	box commands:
\anskey	\box_dp:N 1451, 1452, 1455, 1462, 1475, 1483, 1489,
anskey*	1497, 3984, 3989, 4037, 4128
\anspic	\box_ht:N 1234, 1237, 1248, 1249, 1260, 1262, 1277,
\anspic* 71	1280, 1288, 1289, 1300, 1302, 1317, 1320, 1327, 1328,
\arabic 32, 38	1339, 1341, 1356, 1359, 1405, 1408, 1416, 1417, 1425,
\arabic 487, 613, 659, 5083, 5087, 5103	1426, 1438, 1440
	\box_ht_plus_dp:N 3979, 4045, 4081
В	\box_new:N 70, 149, 150, 183, 189
B base-fix	\box_use_drop:N 4406, 4467, 4747, 5015
base-fix	\box_use_drop:N 4406, 4467, 4747, 5015
base-fix	\box_use_drop:N
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	\box_use_drop:N 4406, 4467, 4747, 5015 \box_wd:N
base-fix 855 \baselineskip 52 \baselineskip $871,878$ before 993 before* 993	\box_use_drop:N
base-fix 855 \baselineskip 52 \baselineskip 871, 878 before 993 before* 993 below 1593	\box_use_drop:N 4406, 4467, 4747, 5015 \box_wd:N
base-fix 855 \baselineskip 52 \baselineskip $871,878$ before 993 before* 993	\box_use_drop:N 4406, 4467, 4747, 5015 \box_wd:N 495 C \c 228, 229, 762, 764, 776, 778 \catcode 2779 \cB 229
base-fix 855 \baselineskip 52 \baselineskip 871,878 before 993 before* 993 below 1593 below* 1593 bool commands:	\box_use_drop:N
base-fix 855 \baselineskip 52 \baselineskip 871,878 before 993 before* 993 below 1593 below* 1593	\box_use_drop:N 4406, 4467, 4747, 5015 \box_wd:N 495 C \c 228, 229, 762, 764, 776, 778 \catcode 2779 \cB 229 \centering 1555, 1582, 4172, 4399, 4460
base-fix 855 \baselineskip 52 \baselineskip 871, 878 before 993 before* 993 below 1593 below* 1593 bool commands: \bool_gset_false:N 357, 358, 359, 2883, 2885, 4411,	\box_use_drop:N
base-fix	\box_use_drop:N
base-fix 855 \baselineskip 52 \baselineskip 871, 878 before 993 before* 993 below 1593 below* 1593 bool commands: \bool_gset_false:N 357, 358, 359, 2883, 2885, 4411, 4415, 4471 \bool_gset_true:N 265, 275, 1096, 2086, 2092, 4380,	\box_use_drop:N
base-fix	\box_use_drop:N
base-fix	\box_use_drop:N 4406, 4467, 4747, 5015 \box_wd:N 495 C \c 228, 229, 762, 764, 776, 778 \catcode 229 \cB 229 \cE 229 \centering 1555, 1582, 4172, 4399, 4460 check-ans 2008 Document class: article 45 clist commands: \clist_const:Nn 195 \clist_map_function:nN 4155 \clist_map_inline:Nn 545, 810, 992, 1007, 1088,
base-fix	\box_use_drop:N
base-fix	\box_use_drop:N 4406, 4467, 4747, 5015 \box_wd:N 495 C \c 228, 229, 762, 764, 776, 778 \catcode 229 \cB 229 \cE 229 \centering 1555, 1582, 4172, 4399, 4460 check-ans 2008 Document class: article 45 clist commands: \clist_const:Nn 195 \clist_map_function:nN 4155 \clist_map_inline:Nn 545, 810, 992, 1007, 1088, 1609 \clist_map_inline:nn 49, 60, 78, 86, 99, 111, 141,
base-fix	\box_use_drop:N

Commands provide by enumext:	\cs_set_eq:NN 4523, 4524, 4715, 4798, 4799, 4979
\anskey 30, 67, 68, 73, 74, 76, 78, 79, 86, 88, 98, 99, 118,	\cs_set_protected:\Nn 931, 947, 960, 972
127, 129, 136	\cs_set_protected:Npn 45, 54, 71, 79, 94, 100, 134,
\anspic* 30, 31, 71, 74, 86, 87, 108, 127, 129	166, 174, 514, 524, 546, 581, 597, 643, 785, 811, 887,
\anspic 30, 74, 106, 108, 136	910, 984, 993, 1072, 1089, 1593, 1704, 1947, 2008,
\foreachkeyans 132, 139	2167, 2209, 2245, 2394, 2936, 3227, 3243, 3283, 3423,
\getkeyans	3464
\item* 30, 31, 71, 74, 86, 87, 89, 90, 93, 119, 120, 125, 127,	\cs_to_str:N 484, 507
129	\cs_undefine:N 2655, 2656, 2657, 2658
\item 89, 93, 112, 119, 121, 124, 125	D
\miniright 29, 50, 58, 59, 100, 139	
\printkeyans* 128	\d
\printkeyans 30, 74, 128, 129	\DeclareDocumentEnvironment 397
\setenumextmeta 131, 139	dim commands:
\setenumext 30, 129-131, 135	\dim_abs:n
Counters defined by enumext:	\dim_add:Nn
enumXiii	\dim_compare:nNnTF . 933, 949, 962, 974, 1252, 1264,
enumXii	1292, 1304, 1331, 1343, 1420, 1428, 1539, 1568, 3393,
enumXiv	3398, 3404, 3410, 3412, 3414, 3569, 3616, 3719, 3736,
enumXi 28, 38	3963, 4194, 4210, 4225, 4241, 4354, 4419
enumXviii	\dim_compare:nTF 2503, 2848, 3658, 3795
enumXvii	\dim_eval:n 871, 4043, 4124
enumXvi	\dim_gset_eq:NN
enumXv	\dim_gzero:N 2887, 4414, 4474
cs commands:	\dim_new:N . 67, 74, 75, 76, 96, 143, 151, 152, 182, 184,
\cs_generate_variant:Nn . 200, 201, 497, 513, 768,	185, 191
784, 2310, 2315, 2391, 2715, 3422, 4157, 5301	\dim_set:Nn 495, 901, 3145, 3396, 3401, 3403, 3406,
\cs_if_exist:NTF 467	3407, 3411, 3413, 3416, 3417, 3419, 3572, 3619, 3657,
\cs_if_free:NTF	3721, 3738, 3794, 3977, 4079, 4160, 4196, 4203, 4227,
\cs_new:\n	4234, 4289, 4338, 4356, 4421, 4663
\cs_new:Npn . 232, 1716, 1725, 1733, 2273, 2282, 2290,	\dim_set_eq:NN 604, 650, 719, 723, 3060, 3061, 3073,
5150, 5159, 5168	3074, 3140, 3434, 3477, 3580, 3746, 4296, 4299, 4300,
\cs_new_eq:NN . 384, 385, 390, 391, 441, 442, 445, 446	4345, 4348, 4349, 4656, 4733, 4997
\cs_new_protected:Nn . 224, 238, 254, 280, 313, 343,	\dim_sub:\Nn 3663, 3800, 4212, 4243
349, 355, 361, 367, 375, 393, 411, 636, 697, 748, 861,	\dim_use:N 934, 942, 1540, 1550, 2381, 2384, 2389, 3160,
1008, 1012, 1016, 1020, 1024, 1028, 1032, 1036, 1040,	3162, 3213, 3570, 3574, 3575, 3577, 3617, 3622, 3623,
1044, 1048, 1052, 1056, 1060, 1064, 1068, 1103, 1115,	3629, 3660, 3665
1148, 1165, 1176, 1193, 1219, 1240, 1365, 1391, 1411,	\dim_zero:N 3469, 3583, 3747, 3990 \dim_zero_new:N 464
1444, 1466, 1501, 1507, 1610, 1624, 1638, 1649, 1660,	\c_zero_dim 936, 950, 963, 975, 1540, 1568, 2505, 2850,
1671, 1682, 1693, 1774, 1877, 1890, 1907, 1928, 1956,	3393, 3398, 3404, 3411, 3570, 3617, 3660, 3719, 3736,
1961, 1986, 2027, 2037, 2080, 2095, 2102, 2111, 2116,	3797, 3963, 4194, 4210, 4225, 4241, 4354, 4419
2121, 2126, 2135, 2140, 2145, 2316, 2340, 2347, 2371,	\dimeval
2378, 2392, 2618, 2637, 2653, 2716, 2752, 2783, 2818,	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
2860, 2881, 2889, 2932, 2947, 2975, 3008, 3044, 3056,	E
3069, 3155, 3165, 3176, 3192, 3208, 3330, 3346, 3362,	\end 2344, 2375, 3601, 3762, 4027, 4174, 5124, 5134, 5142
3376, 3499, 3528, 3557, 3564, 3594, 3611, 3633, 3655,	end internal commands:
3691, 3715, 3732, 3757, 3771, 3792, 3945, 4139, 4153,	\endenumext_mini_page . 1548, 1575, 3644, 3782,
4158, 4182, 4192, 4223, 4352, 4371, 4417, 4436, 4476,	4378, 4442, 4468
4480, 4499, 4533, 4560, 4567, 4576, 4586, 4611, 4765,	\endgroup 2779
4809, 4840, 4846, 4867, 4924, 5033	\endlist 385
\cs_new_protected:Npn 202, 206, 210, 449, 465, 482,	\endminipage
492, 498, 617, 661, 731, 755, 769, 1537, 1566, 1742,	enumext
1761, 1831, 1864, 1966, 2150, 2227, 2237, 2259, 2267,	enumext internal commands:
2302, 2311, 2467, 2530, 2545, 2583, 2587, 2707, 2738,	$local_loc$
2742, 2773, 2909, 2985, 3029, 3109, 3128, 3251, 3255,	\lenumextresume_name_tl 63
3269, 3273, 3291, 3295, 3305, 3317, 3391, 3425, 3466,	\enumext_add_meta_key:nnn 132, 5253, 5269,
3510, 3711, 3954, 3961, 3968, 4070, 4089, 4113, 4254,	5270, 5272, 5275
4303, 4550, 4617, 4624, 4638, 4646, 4651, 4661, 4833,	\enumext_add_pre_parsep: . 51, 1113, 1115, 1115
4873, 4880, 4894, 4902, 4919, 5055, 5068, 5116, 5239,	\enumext_after_args_exec: 48, 1008, 1020, 3682
5251, 5275, 5287, 5325, 5335, 5343, 5365	\enumext_after_args_exec_v: 1024, 1036, 3815
\cs_new_protected_nopar:Nn 3853, 3895, 3903,	\enumext_after_args_exec_vii: 1040, 1064
3911, 4596, 4604, 4741, 4852, 4860, 5009	\enumext_after_args_exec_viii: 1068
\cs_new_protected_nopar:Npn 3845, 3861, 4667,	\enumext_after_env:nn 83-85, 101, 114, 122, 206,
4713, 4952, 4977	206, 2793, 3687, 4387, 4450, 4781
\cs_set:Npn 2402, 2439, 5061	\enumext_after_hyperref: 36, 409, 411, 411
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	~ · · · · · · · · · · · · · · · · · · ·

©2024 by Pablo González L

142 / 156

\lenumext_after_list_args_v_tl 1038
\lenumext_after_list_args_vii_tl 1066, 4739
\lenumext_after_list_args_viii_tl 1070,
5007
\enumext_after_list_vii: 114, 118, 4531, 4567,
4567
\enumext_after_list_viii: 124, 4807, 4846,
4846
\enumext_after_stop_list: 48, 101, 1008, 1016,
3649
\enumext_after_stop_list_v: 1024, 1032, 3789
\lenumext_after_stop_list_v_tl 1034
·
\enumext_after_stop_list_vii: 118, 1040,
1056, 4570
\lenumext_after_stop_list_vii_tl 1058
\enumext_after_stop_list_viii: . 1060, 4849
\lenumext_after_stop_list_viii_tl 1062
\lenumext_align_label_pos_v_str 3380
\lenumext_align_label_pos_X_str <u>79</u>
\lenumext_align_label_vii_str 4702
\lenumext_align_label_viii_str 4966
\lenumext_align_label_X_str <u>174</u>
\cenumext_all_envs_clist 195, 545, 810, 992,
1007, 1088, 1609
** ***
\cenumext_all_families_seq 131, 5207, 5233
\lenumext_anskey_env_bool 33, 82, 34, 290, 305,
2709
\enumext_anskey_env_clean_vars: . 85, 2814,
<u>2818,</u> 2881
\enumext_anskey_env_define_keys: 82, 2707,

2716, 2787
\enumext_anskey_env_exec: 83, 2712, <u>2783</u> , 2783
\enumext_anskey_env_make:n 67, 82, 1991, 2707,
2707 2715
2707, 2715
2707, 2715 \enumext_anskey_env_reset_keys: 83, 84, 2752,
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 _enumext_anskey_env_reset_keys:\enumext_rescan_anskey_env:n 2707
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 _enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 _enumext_anskey_env_save_keys: 84, 2795,
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 _enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 _enumext_anskey_env_save_keys: 84, 2795,
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 _enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 _enumext_anskey_env_store: 84, 2811, 2818, 2860 _enumext_anskey_env_unknown:n 82, 2735, 2738 _enumext_anskey_env_unknown:n . 2740, 2742
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 _enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:n . 2740, 2742 _enumext_anskey_level_int 28, 2639, 2640
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner:n 80
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner: n 80 \enumext_anskey_safe_outer: . 79, 2599, 2618,
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner: n 80 _enumext_anskey_safe_outer: . 79, 2599, 2618, 2618
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner: n 80 \enumext_anskey_safe_outer: . 79, 2599, 2618,
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 _enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \l_enumext_anskey_env_unknown:n 2740, 2742 \l_enumext_anskey_level_int 28, 2639, 2640 _enumext_anskey_safe_inner: 80, 2612, 2618, 2637 _enumext_anskey_safe_inner:n 80 _enumext_anskey_safe_outer: . 79, 2599, 2618, 2618 _enumext_anskey_show_wrap_arg:n 78, 2530,
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner:n 80 \enumext_anskey_safe_outer: . 79, 2599, 2618, 2618 _enumext_anskey_safe_outer:
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 _enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \l_enumext_anskey_env_unknown:n 2740, 2742 \l_enumext_anskey_level_int 28, 2639, 2640 _enumext_anskey_safe_inner: 80, 2612, 2618, 2637 _enumext_anskey_safe_inner:n 80 _enumext_anskey_safe_outer: . 79, 2599, 2618, 2618 _enumext_anskey_show_wrap_arg:n 78, 2530,
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner:n 80 \enumext_anskey_safe_outer: . 79, 2599, 2618, 2618 _enumext_anskey_safe_outer:
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner:n 80 \enumext_anskey_safe_outer: . 79, 2599, 2618, 2618 _enumext_anskey_safe_outer: 79, 2599, 2618, 2618 _enumext_anskey_show_wrap_arg:n 78, 2530, 2530, 2549, 2564 _enumext_anskey_show_wrap_left:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner:n 80 \enumext_anskey_safe_inner:n 80 _enumext_anskey_safe_outer: . 79, 2599, 2618, 2618 _enumext_anskey_show_wrap_arg:n . 78, 2530, 2530, 2549, 2564 _enumext_anskey_show_wrap_left:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n 2707 \enumext_anskey_env_save_keys: 84, 2795, 2818, 2818 \enumext_anskey_env_store: 84, 2811, 2818, 2860 \enumext_anskey_env_unknown:n 82, 2735, 2738 \enumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_env_unknown:nn . 2740, 2742 \lenumext_anskey_level_int 28, 2639, 2640 \enumext_anskey_safe_inner: 80, 2612, 2618, 2637 \enumext_anskey_safe_inner:n 80 \enumext_anskey_safe_inner:n 80 _enumext_anskey_safe_outer: . 79, 2599, 2618, 2618 _enumext_anskey_show_wrap_arg:n . 78, 2530, 2530, 2549, 2564 _enumext_anskey_show_wrap_left:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n
\enumext_anskey_env_reset_keys: 83, 84, 2752, 2815 \enumext_anskey_env_reset_keys:\ enumext_rescan_anskey_env:n

```
\l__enumext_anspic_body_box . . . 142, 4078, 4081
\__enumext_anspic_body_dim:n . . 109, 4051, 4070,
    4116
\l__enumext_anspic_body_htdp_dim .. 109, 142,
    4079, 4127
__enumext_anspic_exec: .......... 4051
\__enumext_anspic_exec: ..... 111, 4022, 4182
\__enumext_anspic_label:nn 109, 4051, 4089, 4119,
    4134
\l__enumext_anspic_label_above_bool . . . 142,
    3923, 3926, 3973, 4035, 4072, 4117, 4144
\l__enumext_anspic_label_box .. 142, 3976, 3979
\l__enumext_anspic_label_htdp_dim . 107, 142,
    3977, 3983, 4045, 4126
\__enumext_anspic_label_pos:nnn .. 110, 4051,
    4113, 4142
\l__enumext_anspic_label_sep_skip 3933, 3985,
    4046, 4129, 4146
\l__enumext_anspic_layout_style_tl 3935, 4184,
\l__enumext_anspic_mini_pos_str .. 142, 3924,
    3927, 4171
\l__enumext_anspic_mini_width_dim
                                      142, 4091,
    4160, 4171
\__enumext_anspic_print:n 110, 4051, 4153, 4157,
    4186, 4189
\__enumext_anspic_row:n . . 110, 4051, 4155, 4158
\__enumext_anspic_start_list_tag: 3869, 3895,
    4141
\__enumext_anspic_stop_list_tag: . 3869, 3911,
\__enumext_anspic_stop_start_list_tag: 3869,
    3903, 4143
\ensuremath{\mbox{\mbox{--}enumext\_at\_begin\_document:n}} .. 35, \underline{202}, 202,
    382, 388
\l__enumext_base_line_fix_bool 45, 46, 129, 857,
    866, 885, 5130, 5135
\__enumext_before_args_exec: 48, 100, 118, 1008,
    1008, 3614
\__enumext_before_args_exec_v: 1024, 1024, 3718
\__enumext_before_args_exec_vii: . 1040, 1040,
    4564
\__enumext_before_args_exec_viii: 1044, 4843
\__enumext_before_env:nn 82, 206, 210, 2660, 2672,
    2684, 2785
\__enumext_before_keys_exec: . . 48, 1008, 1012,
\__enumext_before_keys_exec_v: 1024, 1028, 3812
\__enumext_before_keys_exec_vii ..... 1040
\__enumext_before_keys_exec_vii: . 1048, 4518
\__enumext_before_keys_exec_viii: 1052,4793
\__enumext_before_list: . . 100, 3611, 3611, 3673
\__enumext_before_list_v: ... 3715, 3715, 3807
\__enumext_before_list_vii: . . . 118, 4513, 4560,
    4560
\__enumext_before_list_viii: .. 123, 4789, 4840,
\l__enumext_before_no_starred_key_v_tl 1030
\l__enumext_before_no_starred_key_vii_-
    \l__enumext_before_no_starred_key_viii_-
    tl ..... 1054
\l__enumext_before_starred_key_v_tl . . . 1026
\l__enumext_before_starred_key_vii_tl . 1042
\l__enumext_before_starred_key_viii_tl 1046
```

__enumext_calc_hspace:NNNNNNN 95, 3391, 3391, 3422, 3427, 3470 __enumext_check_ans_active: 68, 100, 118, 2027, 2027, 3615, 4563 \g__enumext_check_ans_item_tl 87 \g__enumext_check_ans_key_bool 69, 70, 153, 357, 2086, 2092, 2899 \l__enumext_check_ans_key_bool 69, 2012, 2017, 2083, 2089 __enumext_check_ans_key_hook: .. 69, 101, 118, <u>2080</u>, 2080, 3650, 4571 __enumext_check_ans_level: . 68, 69, 2027, 2033, __enumext_check_ans_log: 70, 85, 2126, 2126, 2903 __enumext_check_ans_log_msg_greater: 2126, 2132, 2145 __enumext_check_ans_log_msg_less: 2126, 2130, __enumext_check_ans_log_msg_same_ok: 2126, 2131, 2140 __enumext_check_ans_msg_greater: 2102, 2108, __enumext_check_ans_msg_less: 2102, 2106, 2111 $\ensuremath{\mbox{\sc loss}}$ enumext_check_ans_msg_same_ok: 2102, 2107, __enumext_check_ans_show: . . 70, 85, 2102, 2102, 2901 \l__enumext_check_answers_bool 67, 68, 79, 89, 90, 153, 1989, 2016, 2031, 2318, 2342, 2349, 2373, 2601, 2798, 3024, 3113, 3147, 4682 __enumext_check_starred_cmd:n 34, 71, 87, 122, 2150, 2150, 3818, 4033, 4806 \g__enumext_check_starred_cmd_int .. 94, 153, 2153, 2159, 2164, 3328, 4098, 4931 \l__enumext_check_start_line_env_tl . 34, 153, 320, 328, 336, 2156, 2162, 2165 \l__enumext_columns_sep_v_dim 3736, 3738, 3746 \l__enumext_columns_sep_vii_dim .. 4194, 4196, 4205, 4217, 4293, 4762 \l__enumext_columns_sep_viii_dim . 4225, 4227, 4236, 4248, 4342, 5030 \l__enumext_columns_v_int 1385, 1403, 1571, 3734, 3742, 3754, 3759 \l__enumext_columns_vii_int . . 4199, 4202, 4206, 4215, 4257, 4261, 4264, 4270, 4276, 4280, 4756, 4770 \l__enumext_columns_viii_int . 4230, 4233, 4237, 4246, 4306, 4310, 4313, 4319, 4325, 4329, 5024, 5039 \l__enumext_counter_i_tl 45, 474 \l__enumext_counter_ii_tl <u>45</u>, 475 \l__enumext_counter_iii_tl 45, 476 \l__enumext_counter_iv_tl 45, 477 \c__enumext_counter_style_tl 32, 50, 226 \g__enumext_counter_styles_tl . 28, 38, 67, 485, \l__enumext_counter_v_tl 45, 478, 739 \l__enumext_counter_vi_tl 45, 479 \l__enumext_counter_vii_tl 45, 480, 671 \l__enumext_counter_viii_tl 45, 481, 687 \l__enumext_current_widest_dim 28, 67, 509, 605, 651, 720, 724 __enumext_def_meta_key:nnn . . . 132, 5253, 5281, 5287, 5301 __enumext_default_item:n ... 3109, 3109, 3173 $\ensuremath{\mbox{\sc loss}}$ enumext_define_counters:Nn $28,\,465,\,465,\,474,$

```
475, 476, 477, 478, 479, 480, 481
\__enumext_endminipage: . 36, 382, 391, 405, 4408,
    4743, 5011
\g__enumext_envir_name_tl 33, 34, 291, 306, 365,
    1959, 1964, 1974, 2114, 2119, 2124, 2138, 2143, 2148
\l__enumext_envir_name_tl . 33, 34, 34, 260, 270,
    319, 327, 335, 5649, 5652, 5659, 5662, 5669, 5672,
    5679, 5682, 5688, 5692, 5698, 5702, 5759, 5763
\__enumext_execute_after_env: 35, 66, 70, 81, 85,
    2889, 2889, 3689, 4783
\__enumext_fake_item_indent: .. 931, 931, 3454
\l__enumext_fake_item_indent_v_dim 950,955
\l__enumext_fake_item_indent_v_tl 952, 3310,
    3314, 3322
\__enumext_fake_item_indent_vii: .. 931,960,
    3487
\l__enumext_fake_item_indent_vii_dim 963,967
\l__enumext_fake_item_indent_vii_tl 965, 4738
\__enumext_fake_item_indent_viii: . 931,972,
\l__enumext_fake_item_indent_viii_dim . 975,
\l__enumext_fake_item_indent_viii_tl .. 977,
    5002
\l__enumext_fake_item_indent_X_tl .... 100
\__enumext_fake_make_label_vii:n . 120,4667,
    4667, 4730
\__enumext_fake_make_label_viii:n 4952, 4952,
    4994
\__enumext_filter_first_level:n . . 130, 5150,
    5150, 5184, 5195
\__enumext_filter_first_level_key:n 130, 5150,
\__enumext_filter_first_level_pair:nn . 130,
    5150, 5156, 5168
\__enumext_filter_save_key:n . . 73, 2234, 2242,
    2265, 2271, 2273, 2273, 5081, 5085, 5089, 5093, 5097,
    5101
\__enumext_filter_save_key_key:n . . 73, 2273,
    2278, 2282
\__enumext_filter_save_key_pair:nn 74,2273,
    2279, 2290
\__enumext_filter_series:n 62, <u>1716</u>, 1716, 1754,
    1766, 1771
\__enumext_filter_series_key:n 62, 1716, 1721,
\ensuremath{\mbox{\sc loss}} enumext_filter_series_pair:nn .. 62, \underline{1716},
    1722, 1733
\__enumext_first_item_tmp_vii: 116, 118, 4523,
    4596, 4596
\__enumext_first_item_tmp_viii: .. 124, 4798,
    4852, 4852
\g_{\text{enumext\_footnote\_arg\_seq}} . 171, 4482, 4495,
    4505
\verb|\g_enumext_footnote_int| . \underline{171}, 4489, 4492, 4494,
\g__enumext_footnote_int_seq . 171, 4483, 4496,
    4501, 4504
\__enumext_footnotes_key_bool ..... 36
l_enumext_footnotes_key_bool 31, 37, 121, \underline{161},
    420, 427, 436, 4724, 4750, 4988, 5018
\__enumext_footnotetext:nn \dots \underline{4476}, 4476, 4506
\__enumext_foreach_add_body:n . 133, 5302, 5362,
    5365
```

\l__enumext_foreach_after_tl 5306, 5374

\lenumext_foreach_before_tl 5304, 5369
\genumext_foreach_default_keys_tl 132, 126,
5324,5345 \enumext_foreach_keyans:nn 133, <u>5302</u> , 5341,
5343 \lenumext_foreach_name_prop_tl . <u>126</u> , 5347,
537 ² \lenumext_foreach_print_seq <u>126</u> , 5357, 5363,
5367
\lenumext_foreach_sep_tl 5316, 5363
\lenumext_foreach_start_int 5308, 5359 \lenumext_foreach_step_int 5312, 5360
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
5361
\enumext_foreach_wrapper:n 5314, 5370
\enumext_getkeyans:nn 128, 5050, 5064, 5068
\enumext_getkeyans_aux:n 128, 5050, 5052, 5055
\lenumext_hyperref_bool . 31, 36, 37, <u>161</u> , 416, 439, 456, 2520, 3012, 4676
\enumext_hypertarget:nn 37, 411, 441, 445, 461
\enumext_if_is_int:n 218
\enumext_if_is_int:nTF <u>218</u> , 757, 771
\enumext_internal_mini_page: 36, 98, 117, 393,
393, 3501, 4535 \enumext_is_not_nested: 28, 33, 98, 117, 254, 254,
3502, 4536
\enumext_is_on_first_level: . 28, 33, 98, 117,
254, 280, 3508, 4548
\genumext_item_anskey_int 79, 88, 153, 352, 379, 380, 2099, 2469, 3026
\enumext_item_answer_diff: 70, 85, 2095, 2095, 2896
$\g_{\text{enumext_item_answer_diff_int}}$ 70, $\underline{153}$, 353,
2097, 2104, 2128 \lenumext_item_column_pos_vii_int 119, 4264,
4270, 4276, 4280, 4287, 4607, 4756, 4759
\l_enumext_item_column_pos_viii_int 124,
4313, 4319, 4325, 4329, 4336, 4863, 5024, 5027
l_enumext_item_column_pos_X_int 174
\genumext_item_count_all_vii_int 119, 4288, 4608, 4770, 4778
\g_enumext_item_count_all_viii_int 124, 4337,
4864, 5038, 5047
$\g_{\text{enumext_item_count_all_X_int}}$
\g_enumext_item_number_bool 153
\lenumext_item_number_bool 69, 159, 2049, 2054, 2058, 2062, 2075, 2644, 2698, 3116, 3150, 4685
\genumext_item_number_int 69, 153, 351, 378,
380, 2048, 2053, 2057, 2061, 2074, 2099, 3115, 3149,
4684
\enumext_item_peek_args_vii: 119, 4604, 4609,
4611 \enumext_item_peek_args_viii: 124, 4860,
4865, 4867
\enumext_item_star_exec: 90, 3128, 3155, 3200,
3219 \lenumext_item_starred_vii_bool 4626, 4640,
4689 \lenumext_item_starred_viii_bool 4882,4896,
4962, 5003
\lenumext_item_starred_X_bool <u>174</u> \enumext_item_std:w <i>36, 89, 90, 94, 382, 386, 3119,</i>
3125, 3153, 3310, 3314, 3322
\genumext_item_symbol_aux_tl . 90, 130, 3133,

3136, 3161, 3205, 3223

```
\g__enumext_item_symbol_aux_vii_tl 4648, 4691,
    4694, 4698, 4700
\g__enumext_item_symbol_aux_X_tl .... 174
\l__enumext_item_symbol_sep_vii_dim . . 4656,
    4663, 4697, 4699
\l__enumext_item_symbol_vii_tl ..... 4694
\l__enumext_item_text_vii_box .... 4716, 4747
\l__enumext_item_text_viii_box ... 4980, 5015
\l__enumext_item_text_X_box ..... 174
\l__enumext_item_width_vii_dim ... 4203, 4212,
    4291, 4299, 4300
\l__enumext_item_width_viii_dim .. 4234, 4243,
    4340, 4348, 4349
\l__enumext_item_width_X_dim ..... 174
\l__enumext_itemindent_X_dim ..... 71
\l__enumext_itemsep_i_skip . . . 1246, 1253, 1256,
    1258, 1265, 1269, 1272, 1274, 1414, 1421, 1423, 1424,
    1429, 1433, 1435, 1436
\l__enumext_itemsep_ii_skip . . 1286, 1293, 1296,
    1298, 1305, 1309, 1312, 1314
\l__enumext_itemsep_iii_skip . 1325, 1332, 1335,
    1337, 1344, 1348, 1351, 1353
\l__enumext_itemsep_vii_skip ..... 4776
\l__enumext_itemsep_viii_skip ..... 5045
\l__enumext_joined_item_aux_vii_int . . 4285,
    4286, 4287, 4288, 4294
\l__enumext_joined_item_aux_viii_int . 4334,
    4335, 4336, 4337, 4343
\l__enumext_joined_item_aux_X_int .... 174
\__enumext_joined_item_vii:w . . 119, 4604, 4614,
    4615, 4617
\l__enumext_joined_item_vii_int .. 4256, 4257,
    4260, 4262, 4268, 4273, 4278, 4283, 4285, 4291
\__enumext_joined_item_viii:w 124, 4860, 4870,
    4871, 4873
\l__enumext_joined_item_viii_int . 4305, 4306,
    4309, 4311, 4317, 4322, 4327, 4332, 4334, 4340
\l__enumext_joined_item_X_int ..... 174
\l__enumext_joined_width_vii_dim . 4289, 4296,
    4299, 4718, 4732
\l__enumext_joined_width_viii_dim 4338, 4345,
    4348, 4982, 4996
\l__enumext_joined_width_X_dim ..... 174
\__enumext_keyans_addto_prop:n 86, 2909, 2909,
    3325, 4095
\__enumext_keyans_addto_seq:n . 87, 2985, 2985,
    3327, 4097
\__enumext_keyans_addto_seq_link: 2985, 3006,
    3008, 4930
\__enumext_keyans_default_item:n . . 93, 3305,
    3305, 3342
\l__enumext_keyans_env_bool 34, 3533, 3546, 3698,
    3788
\__enumext_keyans_fake_item_indent: 931,947,
    3444
\l__enumext_keyans_level_h_int . . 123, 28, 680,
    706, 2628, 2690, 2963, 4542, 4815, 4816
2686, 2958, 3697, 3702, 4061
\__enumext_keyans_make_label: 39, 94, 3346, 3346,
\__enumext_keyans_make_label_box: 3346, 3350,
    3355, 3376
\__enumext_keyans_make_label_std: 3346, 3358,
```

3362

\enumext_keyans_mini_right_cmd:n

\enumext_keyans_mini_set_vskip: 55
\enumext_keyans_minipage_add_space: 1365,
1391, 3727
\enumext_keyans_minipage_set_skip: . 1365,
1365, 1393
\enumext_keyans_multi_addvspace: 1165, 1176,
3751
\enumext_keyans_multi_set_vskip: 52, 1165,
1165, 1178
\enumext_keyans_multicols_start: 3715, 3730,
3732
\enumext_keyans_multicols_stop: 1570, 3715,

3757, 3786
\enumext_keyans_name_and_start: 28, 34, 123,
313, 313, 3699, 3952, 4820

\enumext_keyans_parse_keys:n <u>3711</u> , 3711, 3806
\enumext_keyans_pic_arg_two: 107, 3945, 3968,
3998
\lenumext_keyans_pic_level_int <u>28</u> , 1515,
2632, 2694, 2912, 2953, 2988, 3076, 3947, 3948
\enumext_keyans_pic_parse_keys:n <u>3945</u> , 3954,
3997
\g_enumext_keyans_pic_parsep_skip 142
$\ensuremath{\mbox{\mbox{-}enumext_keyans_pic_safe_exec:}}$. 106, $\underline{3945}$,
3945, 3996
_enumext_keyans_pic_skip_abs:N . 107, 3945,
3961, 3972
\enumext_keyans_pre_itemsep_skip: 1365,
1384, 1411
\enumext_keyans_redefine_item: 94, 3330,
3330, 3441
\enumext_keyans_ref: $\dots \dots 43, \underline{731}, 748, 3443$
\enumext_keyans_ref:n 43, 728, 731, 731
\enumext_keyans_safe_exec: . 3691, 3691, 3805
\enumext_keyans_set_item_width: $103, \frac{3792}{2}$,
3702. 3814
3792, 3814
$\verb \enumext_keyans_show_ans: \underline{3029}, 3037, 3056$
$\verb \enumext_keyans_show_ans: \underline{3029}, 3037, 3056$
$\label{lem:condition} $$\cline{2.5}$ enumext_keyans_show_ans: $$\underline{3029}$, 3037, 3056 \\ $\cline{2.5}$ enumext_keyans_show_item_opt: $$94$, $$\underline{3029}$, $$$
\enumext_keyans_show_ans: $\underline{3029}$, 3037 , 3056 \enumext_keyans_show_item_opt: 94 , $\underline{3029}$, 3044 , 3323 , 4110 , 5005
$\label{lem:condition} $$\cline{2.5}$ enumext_keyans_show_ans: $$\underline{3029}$, 3037, 3056 \\ $\cline{2.5}$ enumext_keyans_show_item_opt: $$94$, $$\underline{3029}$, $$$
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: 94, <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . 94, <u>3029</u> , 3029,
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: <i>94</i> , <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . <i>94</i> , <u>3029</u> , 3029, 3320, 4104
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: <i>94</i> , <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . <i>94</i> , <u>3029</u> , 3029, 3320, 4104 \enumext_keyans_show_pos: <u>3029</u> , 3041, 3069
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: <i>94</i> , <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . <i>94</i> , <u>3029</u> , 3029, 3320, 4104
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: 94, <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . 94, <u>3029</u> , 3029, 3320, 4104 \enumext_keyans_show_pos: <u>3029</u> , 3041, 3069 \enumext_keyans_starred_item:n 94, <u>3317</u> ,
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: 94, <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . 94, <u>3029</u> , 3029, 3320, 4104 \enumext_keyans_show_pos: <u>3029</u> , 3041, 3069 \enumext_keyans_starred_item:n 94, <u>3317</u> , 3317, 3338
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: 94, <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . 94, <u>3029</u> , 3029, 3320, 4104 \enumext_keyans_show_pos: <u>3029</u> , 3041, 3069 \enumext_keyans_starred_item:n 94, <u>3317</u> , 3317, 3338 \enumext_keyans_store_ref: 86, <u>2932</u> , 2932,
\enumext_keyans_show_ans: <u>3029</u> , 3037, 3056 \enumext_keyans_show_item_opt: 94, <u>3029</u> , 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . 94, <u>3029</u> , 3029, 3320, 4104 \enumext_keyans_show_pos: <u>3029</u> , 3041, 3069 \enumext_keyans_starred_item:n 94, <u>3317</u> , 3317, 3338 \enumext_keyans_store_ref: 86, <u>2932</u> , 2932,
\enumext_keyans_show_ans: 3029, 3037, 3056 \enumext_keyans_show_item_opt: 94, 3029,
\enumext_keyans_show_ans: 3029, 3037, 3056 \enumext_keyans_show_item_opt: 94, 3029, 3044, 3323, 4110, 5005 \enumext_keyans_show_left:n . 94, 3029, 3029, 3320, 4104 \enumext_keyans_show_pos: 3029, 3041, 3069 \enumext_keyans_starred_item:n 94, 3317, 3317, 3338 \enumext_keyans_store_ref: 86, 2932, 2932, 3326, 4096, 4928 \enumext_keyans_store_ref_aux_i: 86, 2932,
\enumext_keyans_show_ans: 3029, 3037, 3056 \enumext_keyans_show_item_opt: 94, 3029,

```
\l__enumext_label_fill_left_v_tl .... 3366
\l__enumext_label_fill_left_X_tl ..... 100
\l__enumext_label_fill_right_v_tl .... 3373
\l__enumext_label_fill_right_X_tl .... <u>100</u>
\l__enumext_label_font_style_v_tl 3367, 3382,
\l__enumext_label_font_style_vii_tl . . . 4704
\l__enumext_label_font_style_viii_tl .. 4968
\l__enumext_label_i_tl ..... 597
\l__enumext_label_ii_tl ..... 597
\l__enumext_label_iii_tl ......
\l__enumext_label_iv_tl ..... 597
\__enumext_label_style:Nnn 28, 38, 498, 498, 513,
    602, 648, 717, 721
\l__enumext_label_v_tl 87,714,2917,2993,3063,
    3103, 3319, 3324, 3809, 3976, 4103, 4105
\l__enumext_label_vi_tl 87, 714, 2914, 2990, 4103,
    4105, 4109
\l__enumext_label_vii_tl . 643, 4635, 4658, 4665
\l__enumext_label_viii_tl 643, 4891, 4922, 4926
\l__enumext_label_width_by_box . . 67, 494, 495
\__enumext_label_width_by_box:Nn 38,492,492,
    497, 509, 781
\l__enumext_labelsep_i_dim . . . 3061, 3066, 3074,
    3106, 4934, 4949
\l__enumext_labelsep_v_dim ..... 3741
\l__enumext_labelsep_vii_dim . 2536, 3061, 3074,
    4198, 4208, 4292, 4600, 4656, 4711, 4720
\l__enumext_labelsep_viii_dim 4229, 4239, 4341,
    4856, 4975, 4984
\l__enumext_labelwidth_i_dim . 3060, 3066, 3073,
    3106, 4934, 4949
\l__enumext_labelwidth_v_dim .... 3380, 3741
\l__enumext_labelwidth_vii_dim ... 2536, 3060,
    3073, 4198, 4207, 4292, 4600, 4702, 4719
\l__enumext_labelwidth_viii_dim .. 4229, 4238,
    4341, 4856, 4966, 4983
\l__enumext_leftmargin_tmp_v_bool . 107, 3970
\l__enumext_leftmargin_tmp_X_bool ..... 71
\l__enumext_leftmargin_tmp_X_dim ..... 71
\l__enumext_leftmargin_X_dim ..... 71
\__enumext_level: 214, 214, 626, 629, 630, 638, 640,
    934, 938, 942, 1010, 1014, 1018, 1022, 1105, 1107,
    1109, 1111, 1153, 1155, 1157, 1159, 1163, 1197, 1203,
    1208, 1210, 1213, 1216, 1229, 1232, 1540, 1544, 1550,
    1613, 1615, 1617, 1620, 1627, 1629, 1631, 1634, 2229,
    2231, 2233, 2261, 2262, 2264, 2320, 2328, 2332, 2336,
    2540, 2541, 3118, 3119, 3123, 3124, 3125, 3133, 3141,
    3142, 3145, 3152, 3153, 3157, 3160, 3162, 3196, 3197,
    3198, 3201, 3204, 3213, 3214, 3216, 3217, 3220, 3539,
    3552, 3559, 3567, 3570, 3572, 3574, 3575, 3576, 3577,
    3580, 3585, 3591, 3597, 3604, 3617, 3619, 3622, 3623,
    3625, 3629, 3635, 3660, 3665, 3676, 3678
\l__enumext_level_h_int 117, 28, 263, 286, 300, 664,
    699, 1522, 2045, 2065, 2430, 2664, 2676, 3547, 4537,
    4538
\l__enumext_level_int . 98, 28, 216, 273, 285, 301,
    395, 1117, 1242, 1521, 2039, 2071, 2407, 2417, 2423,
    2429, 2436, 2445, 2450, 2663, 2675, 2891, 3458, 3503,
    3504, 3515, 3523, 3537, 3550, 3581, 3706, 4057, 4580,
    4590, 4828, 5689, 5693, 5699, 5703
\__enumext_list_arg_two_i: ..... 3423
\__enumext_list_arg_two_ii: ..... 3423
\__enumext_list_arg_two_iii: ..... 3423
```

\enumext_list_arg_two_iv: 3423
\enumext_list_arg_two_v: . 94, 3423, 3811, 3971
\enumext_list_arg_two_vii: 3464, 4517
\enumext_list_arg_two_viii: 3464, 4792
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\lenumext_listparindent_vii_dim 4733, 4737
\lenumext_listparindent_viii_dim 4997, 5001
\enumext_log_answer_vars: . 35, 367, 375, 2898
\enumext_log_global_vars: . 35, <u>367</u> , 367, 2897
\enumext_make_label: . 39, 91, 3176, 3176, 3452
\enumext_make_label_box: 3176, 3180, 3185,
3208
\enumext_make_label_std: 3176, 3188, 3192
\lenumext_mark_answer_sym_tl 75, 2183, 2386,
2553, 3078, 3091, 4938
\lenumext_mark_position_str $\underline{130}$, 2187, 2188,
2214, 2215, 2384
\lenumext_mark_ref_sym_tl 2200, 2525, 3020
\lenumext_meta_path_tl . <u>126</u> , 5277, 5278, 5280,
5281
$\c_enumext_meta_paths_prop \dots 131, \underline{5253}$
\enumext_mini_addvspace_vii: 57, 1501, 1501,
4366
\enumext_mini_addvspace_viii: 57, 1501, 1507,
4431
enumext_mini_env* 393
_enumext_mini_page 1550, 1577, 3629, 3728, 4368,
4433, 4454
\enumext_mini_right_cmd:n . 58, 59, 1535, <u>1537</u> ,
1537
\enumext_mini_set_vskip_vii: 56, 1444, 1444,
1503
\enumext_mini_set_vskip_viii: 56, 1444, 1466,
1509
\enumext_minipage:w 36, 382, 390, 399, 4391, 4732,
4996
\lenumext_minipage_active_v_bool 3725, 3748,
3773
\genumext_minipage_active_vii_bool 114,
4380, 4389, 4411
\lenumext_minipage_active_vii_bool . 4362,
4373
\genumext_minipage_active_viii_bool 4444,
4452, 4471
\lenumext_minipage_active_viii_bool 4427,
4438
\genumext_minipage_active_X_bool 174
\lenumext_minipage_active_X_bool 87

\enumext_minipage_add_space: . 53 , 100 , $\underline{1193}$,
\enumext_minipage_add_space: . 53, 100, 1193, 1219, 3627
\enumext_minipage_add_space: . 53 , 100 , 1193 , 1219 , 3627 \genumext_minipage_after_skip 87 , 1448 , 1460 ,
\enumext_minipage_add_space: . 53 , 100 , 1193 , 1219 , 3627 \genumext_minipage_after_skip 87 , 1448 , 1460 , 4409 , 4469
\enumext_minipage_add_space: $.53, 100, \underline{1193}, 1219, 3627$ \genumext_minipage_after_skip $\underline{87}, 1448, 1460, 4409, 4469$ \lenumext_minipage_after_skip $52, 100, \underline{87},$
\enumext_minipage_add_space: . 53, 100, 1193, 1219, 3627 \genumext_minipage_after_skip 87, 1448, 1460, 4409, 4469 \lenumext_minipage_after_skip 52, 100, 87, 1206, 1246, 1248, 1253, 1256, 1260, 1265, 1269, 1272,
\enumext_minipage_add_space: $.53, 100, \underline{1193}, 1219, 3627$ \genumext_minipage_after_skip $.52, 1448, 1460, 4409, 4469$ \lenumext_minipage_after_skip $$
\enumext_minipage_add_space: . 53, 100, 1193, 1219, 3627 \genumext_minipage_after_skip 87, 1448, 1460, 4409, 4469 \lenumext_minipage_after_skip 52, 100, 87, 1206, 1246, 1248, 1253, 1256, 1260, 1265, 1269, 1272,
\enumext_minipage_add_space: $.53, 100, \underline{1193}, 1219, 3627$ \genumext_minipage_after_skip $.52, 1448, 1460, 4409, 4469$ \lenumext_minipage_after_skip $$
\enumext_minipage_add_space: $.53, 100, \underline{1193}, 1219, 3627$ \genumext_minipage_after_skip $.52, 100, \underline{87}, 1448, 1460, 4409, 4469$ \lenumext_minipage_after_skip $$
\enumext_minipage_add_space: $.53, 100, \underline{1193}, 1219, 3627$ \genumext_minipage_after_skip $.52, 100, \underline{1193}, 4409, 4409, 4469$ \\l_enumext_minipage_after_skip $$
\enumext_minipage_add_space: $.53, 100, \underline{1193}, 1219, 3627$ \genumext_minipage_after_skip $.52, 100, \underline{87}, 1448, 1460, 4409, 4469$ \l_enumext_minipage_after_skip $$
\enumext_minipage_add_space: . 53, 100, 1193, 1219, 3627 \genumext_minipage_after_skip 87, 1448, 1460, 4409, 4469 \lenumext_minipage_after_skip 52, 100, 87, 1206, 1246, 1248, 1253, 1256, 1260, 1265, 1269, 1272, 1276, 1288, 1293, 1296, 1300, 1305, 1309, 1312, 1316, 1327, 1332, 1335, 1339, 1344, 1348, 1351, 1355, 1367, 1381, 1414, 1416, 1421, 1423, 1425, 1429, 1433, 1435, 1437, 1468, 1481, 1495, 1546, 1573, 3783 \genumext_minipage_center_vii_bool . 4395,
\enumext_minipage_add_space: . 53, 100, 1193, 1219, 3627 \genumext_minipage_after_skip 87, 1448, 1460, 4409, 4469 \lenumext_minipage_after_skip 52, 100, 87, 1206, 1246, 1248, 1253, 1256, 1260, 1265, 1269, 1272, 1276, 1288, 1293, 1296, 1300, 1305, 1309, 1312, 1316, 1327, 1332, 1335, 1339, 1344, 1348, 1351, 1355, 1367, 1381, 1414, 1416, 1421, 1423, 1425, 1429, 1433, 1435, 1437, 1468, 1481, 1495, 1546, 1573, 3783 \genumext_minipage_center_vii_bool . 4395, 4412
\enumext_minipage_add_space: . 53, 100, 1193, 1219, 3627 \genumext_minipage_after_skip 87, 1448, 1460, 4409, 4469 \lenumext_minipage_after_skip 52, 100, 87, 1206, 1246, 1248, 1253, 1256, 1260, 1265, 1269, 1272, 1276, 1288, 1293, 1296, 1300, 1305, 1309, 1312, 1316, 1327, 1332, 1335, 1339, 1344, 1348, 1351, 1355, 1367, 1381, 1414, 1416, 1421, 1423, 1425, 1429, 1433, 1435, 1437, 1468, 1481, 1495, 1546, 1573, 3783 \genumext_minipage_center_vii_bool . 4395, 4412 \genumext_minipage_center_viii_bool 4456,

```
\l__enumext_minipage_hsep_vii_dim .... 4360
\l__enumext_minipage_hsep_viii_dim ... 4425
\l__enumext_minipage_left_skip 87, 1368, 1446,
    1451, 1455, 1469, 1473, 1487, 1505, 1511
\l__enumext_minipage_left_v_dim .. 3721, 3728
\l__enumext_minipage_left_vii_dim 4356, 4368
\l__enumext_minipage_left_viii_dim 4421, 4433
\l__enumext_minipage_left_X_dim ..... 87
\g__enumext_minipage_right_skip 87, 1447, 1452,
    1456, 4394, 4455
\l__enumext_minipage_right_skip . 52, 87, 1195,
    1201,\, 1206,\, 1208,\, 1210,\, 1369,\, 1370,\, 1376,\, 1381,\, 1382,\,
    1383, 1388, 1470, 1477, 1491, 1552, 1579
\l__enumext_minipage_right_v_dim . 1568, 1577,
    3719, 3723
\g__enumext_minipage_right_vii_dim 114, 4364,
    4391, 4414
\l__enumext_minipage_right_vii_dim 114,4354,
    4359, 4365
\g__enumext_minipage_right_viii_dim . . 4429,
    4454, 4474
\l__enumext_minipage_right_viii_dim . . 4419,
    4424, 4430
\g__enumext_minipage_right_X_dim .... 174
\g__enumext_minipage_right_X_skip .... 174
\__enumext_minipage_set_skip: . 52, 1193, 1193,
    1221
\g__enumext_minipage_stat_int . . 100, 87, 1557,
    1584, 3626, 3637, 3642, 3726, 3775, 3780
\l__enumext_minipage_temp_skip 87, 1267, 1277,
    1280, 1307, 1317, 1320, 1346, 1356, 1359, 1431, 1438,
\l__enumext_miniright_code_vii_box 4402,4406
\g__enumext_miniright_code_vii_tl 114, 4397,
    4404, 4413
\l__enumext_miniright_code_viii_box . . 4463,
    4467
\g__enumext_miniright_code_viii_tl 4458, 4465,
\l__enumext_miniright_code_X_box .... 174
\l__enumext_mode_box_bool .... 518, 3183, 3353
\__enumext_multi_addvspace: . 51, 99, <u>1148</u>, 1148,
    3588
\__enumext_multi_set_vskip: 50, 1103, 1103, 1150
\l__enumext_multicols_above_ii_skip . . . 1122
\l__enumext_multicols_above_iii_skip . . 1131
\l__enumext_multicols_above_iv_skip . . . 1140
\l__enumext_multicols_above_v_skip 1167, 1181,
    1191, 1382
\l__enumext_multicols_above_X_skip .... 79
\l__enumext_multicols_below_ii_skip . . 1249,
    1258, 1262, 1274, 1279
\l__enumext_multicols_below_iii_skip . 1289,
    1298, 1302, 1314, 1319
\l__enumext_multicols_below_iv_skip . . 1328,
    1337, 1341, 1353, 1358
\l__enumext_multicols_below_v_skip 1171, 1185,
    1383, 1417, 1424, 1426, 1436, 1439, 3765
\l__enumext_multicols_below_X_skip .... 79
\g__enumext_multicols_right_X_skip .... 79
\__enumext_multicols_start: 99, 100, 3564, 3564,
\__enumext_multicols_stop: 100, 1542, 3594, 3594,
    3647
```

\enumext_nested_base_line_fix: . 45, 98, 855,
861, 3519
\enumext_newlabel:nn 31, 37, 77, 449, 449, 2461,
2979
\lenumext_newlabel_arg_one_tl 31, 37, 77, 86,
163, 2454, 2462, 2524, 2968, 2980, 3018

\l_enumext_newlabel_arg_two_tl 31, 37, 76, 163,
2410, 2420, 2433, 2448, 2463, 2955, 2960, 2965, 2981
\enumext_parse_foreach_keys:n 5302, 5318,
5335
\enumext_parse_foreach_keys:nn . 5302, 5325,

5337
\enumext_parse_keys:n 45, 62, <u>3510</u> , 3510, 3672
\enumext_parse_keys_vii:n 62, 4512, 4550, 4550
\enumext_parse_keys_viii:n . 4788, 4833, 4833
_enumext_parse_save_key:n 73, 2254, 2259, 2259

\enumext_parse_save_key_vii:n 73 , 2249, 2259 ,
2267
\enumext_parse_series:n 62, 98, 117, 1742, 1742,
3518, 4556
\enumext_parse_store_keys:n 98
\\\enumext_parsep_i_skip 1120, 1124
\lenumext_parsep_ii_skip 1129, 1133
\lenumext_parsep_iii_skip 1138, 1142
\lenumext_parsep_vii_skip 4734
\lenumext_parsep_viii_skip 4998
\lenumext_partopsep_v_skip . 1183, 1187, 1378,
1401
\lenumext_partopsep_viii_skip 1479
\enumext_phantomsection: $37, \underline{411}, 442, 446, 462$
\enumext_pre_itemsep_skip: 53, 1211, 1240,
1240
\enumext_print_footnote: 4476, 4499, 4752,

5020
\enumext_print_keyans_box:NN 75 , $\underline{2378}$, 2378 ,
2391, 2535, 2539, 3065, 3105, 4934, 4949
\lenumext_print_keyans_i_tl 5086, 5108
\lenumext_print_keyans_ii_tl 5090, 5109
\lenumext_print_keyans_iii_tl 5094, 5110
\lenumext_print_keyans_iii_tl 5094, 5110 \lenumext_print_keyans_iv_tl 5098, 5111
\lenumext_print_keyans_iii_tl 5094, 5110
\lenumext_print_keyans_iii_tl 5094, 5110 \lenumext_print_keyans_iv_tl 5098, 5111
\lenumext_print_keyans_iii_tl 5094, 5110 \lenumext_print_keyans_iv_tl 5098, 5111 \lenumext_print_keyans_star_bool 45, 46, 129, 130, 867, 875, 5131, 5136
\\\enumext_print_keyans_iii_tl 5094, 5110 \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129, \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 _enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129, 130, 867, 875, 5131, 5136 \\\enumext_print_keyans_starred_tl 128, 129, 130, 5082, 5129 \\\enumext_print_keyans_vii_tl 128, 5102, 5112 \\\enumext_print_keyans_vii_tl 128, 5102, 5112 \\\enumext_print_keyans_X_tl 130 \\enumext_printkeyans:nnn 129, 5105, 5113, 5116 \\enumext_redefine_item: .90, 3165, 3165, 3451 \\\enumext_ref_key_arg_tl 41, 50, 229, 619, 620, 632, 663, 666, 676, 682, 692, 733, 734, 744 \\\enumext_ref_the_count_tl . 41, 50, 626, 629, 632, 671, 673, 676, 687, 689, 692, 739, 741, 744 \\enumext_regex_counter_style: 32, 41, 224, 224, 627, 672, 688, 740
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 _enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129, 130, 867, 875, 5131, 5136 \\\enumext_print_keyans_starred_tl 128, 129, 130, 5082, 5129 \\\enumext_print_keyans_vii_tl 128, 5102, 5112 \\\enumext_print_keyans_Vii_tl 128, 5102, 5112 \\\enumext_print_keyans_X_tl 130 \\enumext_printkeyans:nnn 129, 5105, 5113, 5116 \\enumext_redefine_item: 90, 3165, 3165, 3451 \\\enumext_ref_key_arg_tl 41, 50, 229, 619, 620, 632, 663, 666, 676, 682, 692, 733, 734, 744 \\\enumext_ref_the_count_tl . 41, 50, 626, 629, 632, 671, 673, 676, 687, 689, 692, 739, 741, 744 \\enumext_regex_counter_style: 32, 41, 224, 224, 627, 672, 688, 740 \\enumext_register_counter_style:Nn 482, 482, 487, 488, 489, 490, 491 \\enumext_remove_extra_parsep_vii: 4530, 4765, 4765 \\enumext_remove_extra_parsep_viii: 4805,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129, 130, 867, 875, 5131, 5136 \\\enumext_print_keyans_starred_tl 128, 129, 130, 5082, 5129 \\\enumext_print_keyans_vii_tl 128, 5102, 5112 \\\enumext_print_keyans_Vii_tl 128, 5102, 5112 \\\enumext_print_keyans_X_tl 130 \\enumext_printkeyans:nnn 129, 5105, 5113, 5116 \\enumext_redefine_item: 90, 3165, 3165, 3451 \\\enumext_ref_key_arg_tl 41, 50, 229, 619, 620, 632, 663, 666, 676, 682, 692, 733, 734, 744 \\\enumext_ref_the_count_tl . 41, 50, 626, 629, 632, 671, 673, 676, 687, 689, 692, 739, 741, 744 \\enumext_regex_counter_style: 32, 41, 224, 224, 627, 672, 688, 740 \\enumext_register_counter_style:Nn 482, 482, 487, 488, 489, 490, 491 \\enumext_remove_extra_parsep_vii: 4530, 4765, 4765 \\enumext_remove_extra_parsep_viii: 4805,
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129, 130, 867, 875, 5131, 5136 \\\enumext_print_keyans_starred_tl 128, 129, 130, 5082, 5129 \\\enumext_print_keyans_vii_tl 128, 5102, 5112 \\\enumext_print_keyans_Vii_tl 128, 5102, 5112 \\\enumext_print_keyans_X_tl 130 \\enumext_printkeyans:nnn 129, 5105, 5113, 5116 \\enumext_redefine_item: 90, 3165, 3165, 3451 \\\enumext_ref_key_arg_tl 41, 50, 229, 619, 620, 632, 663, 666, 676, 682, 692, 733, 734, 744 \\\\enumext_ref_the_count_tl 41, 50, 626, 629, 632, 671, 673, 676, 687, 689, 692, 739, 741, 744 \\enumext_regex_counter_style: 32, 41, 224, 224, 627, 672, 688, 740 \\enumext_register_counter_style: Nn 482, 482, 487, 488, 489, 490, 491 \\enumext_remove_extra_parsep_vii: 4530, 4765, 4765 _enumext_remove_extra_parsep_viii: 4805, 5033, 5033 _enumext_renew_footnote: 4476, 4480, 4726, 4990
\\enumext_print_keyans_iii_tl 5094, 5110 \\\enumext_print_keyans_iv_tl 5098, 5111 \\\enumext_print_keyans_star_bool 45, 46, 129, 130, 867, 875, 5131, 5136 \\\enumext_print_keyans_starred_tl 128, 129, 130, 5082, 5129 \\\enumext_print_keyans_vii_tl 128, 5102, 5112 \\\enumext_print_keyans_Vii_tl 128, 5102, 5112 \\\enumext_print_keyans_X_tl 130 \\enumext_printkeyans:nnn 129, 5105, 5113, 5116 \\enumext_redefine_item: 90, 3165, 3165, 3451 \\\enumext_ref_key_arg_tl 41, 50, 229, 619, 620, 632, 663, 666, 676, 682, 692, 733, 734, 744 \\\\enumext_ref_the_count_tl 41, 50, 626, 629, 632, 671, 673, 676, 687, 689, 692, 739, 741, 744 \\enumext_regex_counter_style: 32, 41, 224, 224, 627, 672, 688, 740 \\enumext_register_counter_style: Nn 482, 482, 487, 488, 489, 490, 491 \\enumext_remove_extra_parsep_vii: 4530, 4765, 4765 \\enumext_remove_extra_parsep_viii: 4805, 5033, 5033 \\enumext_renew_footnote: 4476, 4480, 4726,

```
\l__enumext_renew_the_count_viii_tl 690, 708,
            710
\l__enumext_renew_the_count_X_tl ..... 50
\__enumext_rescan_anskey_env:n . . 83, 84, 2773,
            2868, 2876
\__enumext_reset_global_bool: . . 343, 346, 355
\__enumext_reset_global_int: ... 343, 345, 349
\__enumext_reset_global_tl: .... 343, 347, 361
\__enumext_reset_global_vars: . 35, 85, 343, 343,
            2906
\label{locality} $$ l\_enumext\_resume\_active\_bool $62, 64, \underline{61}, 1746, $
            1866
\__enumext_resume_counter: . . 64, 65, 1864, 1870,
            1877
\__enumext_resume_counter:n . 62, 64, 1835, 1840,
            1864, 1864, 1934, 1942
\__enumext_resume_counter_save_ans: 65, 1864,
            1875, 1907
\__enumext_resume_counter_series: . 65, 1864,
            1873, 1890
\g__enumext_resume_int ... <u>61</u>, 1787, 1881, 1882
\__enumext_resume_last:n 62, 63, 1742, 1748, 1761
\l__enumext_resume_name_tl 61, 1783, 1791, 1794,
            1810, 1818, 1821, 1867, 1868, 1896, 1903
\__enumext_resume_save_counter: . 63, 101, 118,
            1774, 1774, 3653, 4574
\__enumext_resume_series:n . 64, 1710, 1831, 1831
\__enumext_resume_starred: . 66, 1711, 1928, 1928
\g__enumext_resume_vii_int 61, 1814, 1886, 1887
\l__enumext_rightmargin_vii_dim . . 4210, 4214,
            4219
\l__enumext_rightmargin_viii_dim . 4241, 4245,
            4250
\__enumext_safe_exec: . . 36, 98, <u>3499</u>, 3499, 3671
\__enumext_safe_exec_vii: . 36, 4511, 4533, 4533
\__enumext_safe_exec_viii: 123, 4787, 4809, 4809
\__enumext_second_part: .. 100, 3633, 3633, 3685
\__enumext_second_part_v: ... 3715, 3771, 3819
\l__enumext_series_name_tl .... 64, 65
\l__enumext_series_str . . 63, 98, 117, 1708, 1744,
            1752, 1753, 1755, 1757, 1778, 1781, 1785, 1805, 1808,
            1812, 3514, 4554
\__enumext_set_error:nn .... 5212, 5249, 5251
\__enumext_set_item_width: 101, 3655, 3655, 3681
\__enumext_set_parse:n ..... 5212, 5223, 5239
\l__enumext_setkey_tmpa_int . . . <u>121</u>, 5216, 5220
\label{local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_loc
            5230, 5232, 5234, 5246
\l__enumext_setkey_tmpa_tl . . . . <u>121</u>, 5222, 5226
\label{local_enumext_setkey_tmpb_seq} 1. 121,5215,5218,
            5222, 5223
\label{local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_loc
\l__enumext_show_answer_bool . 2194, 2218, 2547,
            3035, 3049, 4100, 4932
\__enumext_show_length:nnn . . 48, 232, 232, 5460,
            5461, 5462, 5463, 5464, 5465, 5466, 5467, 5468, 5469,
            5475, 5476, 5477, 5478, 5479, 5480, 5481, 5482, 5483,
            5484
\l__enumext_show_position_bool ... 2197, 2221,
            2551, 3039, 3050, 4101, 4936
\g__enumext_standar_bool 33, 98, 34, 262, 265, 284,
            358, 1776, 1841, 1853, 1879, 1892, 1930, 2070, 2084,
```

©2024 by Pablo González L

2415, 2428, 2443, 3534

- \l__enumext_standar_bool 98, 101, 34, 2416, 3506, 3652, 4547 \l__enumext_standar_first_bool 33, 98, 34, 289, 1763, 1910, 1972, 1979 __enumext_standar_item_vii:w . 119, 4604, 4622, __enumext_standar_item_viii:w 124, 125, 4860, 4878, 4880 __enumext_standar_ref: 41,617,636,3453 __enumext_standar_ref:n 41, 609, 617, 617 $\g_{\text{enumext_standar_series_tl}}$. $\underline{61}$, 1765, 1766, 1932, 1935 __enumext_standar_unknown_keys:n 3283, 3287, __enumext_standar_unknown_keys:nn 3283, 3293, \g__enumext_starred_bool 33, 117, 34, 272, 275, 299, 359, 1803, 1846, 1857, 1884, 1899, 1938, 2044, 2090, 2406, 2949, 4415 \l__enumext_starred_bool 117, 118, 123, 34, 1527, 2444, 2479, 2485, 2533, 2822, 2827, 3058, 3071, 3507, 4546, 4573, 4821, 4825__enumext_starred_columns_set_vii: . . 4192, 4192, 4521
- __enumext_starred_columns_set_viii: . 4192, 4223, 4796 \l__enumext_starred_first_bool 33, 117, 34, 304, 865, 874, 1768, 1919, 1972, 1979 __enumext_starred_item:nn . . . 3128, 3128, 3171 __enumext_starred_item_exec: . 125, 4894, 4924, __enumext_starred_item_vii:w . 119, 4604, 4621, 4638 __enumext_starred_item_vii_aux_i:w . . 4604, 4643, 4646 __enumext_starred_item_vii_aux_ii:w . 4604, 4644, 4649, 4651 __enumext_starred_item_vii_aux_iii:w 4604, 4654, 4661 __enumext_starred_item_viii:w 124, 125, 4877, 4894, 4894 __enumext_starred_item_viii_aux_i:w . . 125, 4894, 4899, 4902 __enumext_starred_item_viii_aux_ii:w . 125, 4894, 4900, 4917, 4919 __enumext_starred_joined_item_vii:n 112, 119, 4254, 4254, 4619 __enumext_starred_joined_item_viii:n . 112, 124, 4254, 4303, 4875 __enumext_starred_ref: 42, <u>661</u>, 697, 3484 __enumext_starred_ref:n 42, 655, 661, 661 $\g_{\text{enumext_starred_series_tl}}$. $\underline{61}$, 1770, 1771, 1940, 1943 __enumext_starred_unknown_keys:n 3265, 3267, 3269 __enumext_starred_unknown_keys:nn 3265, 3271, __enumext_start_from:NNn 43, 755, 755, 768, 790, \l__enumext_start_i_int 1882, 1894, 1913 __enumext_start_item_tmp_vii: 116, 4524, 4604, __enumext_start_item_tmp_viii: ...4799, 4860,
- __enumext_start_item_vii:w 119, 121, 4630, 4635, 4658, 4665, 4713, 4713 __enumext_start_item_viii:w . . 125, 4886, 4891, 4922, 4977, 4977 \g__enumext_start_line_tl 33, 34, 292, 307, 364, 2114, 2119, 2124, 2138, 2143, 2148 __enumext_start_list:nn . 36, 95, 382, 384, 3675, 3808, 4515, 4790 __enumext_start_list_tag:n . . 3821, 3845, 4729, __enumext_start_mini_vii: 118, 4352, 4352, 4565 __enumext_start_mini_viii: ... 123, 4417, 4417, __enumext_start_save_ans_msg: . . 66, 67, 1956, 1956, 1981 __enumext_start_store_level: . 98, 3528, 3528, __enumext_start_store_level_vii: 118,4514, 4576, 4576 \l__enumext_start_vii_int ... 1887, 1901, 1922 \l__enumext_start_X_int <u>100</u> __enumext_stop_item_tmp_vii: 116, 118, 119, 121, 4523, 4529, 4606, 4715 __enumext_stop_item_tmp_viii: 124, 4798, 4804, 4862, 4979 __enumext_stop_item_vii: 121, 4713, 4715, 4741 __enumext_stop_item_viii: . . . 4977, 4979, 5009 __enumext_stop_list: 36, 114, 118, 382, 385, 3599, 3607, 3761, 3768, 4375, 4383, 4440, 4447 __enumext_stop_list_tag:n . . . 3821, 3861, 4744, __enumext_stop_mini_vii: 114, 118, 4352, 4371, 4569 __enumext_stop_mini_viii: 124, 4417, 4436, 4848 __enumext_stop_save_ans_msg: . 66, 1956, 1961, __enumext_stop_start_list_tag: .. 3821, 3853, 4731, 4995 __enumext_stop_store_level: 99, 100, 3557, 3557, 3600, 3608 __enumext_stop_store_level_vii: .. 114, 118, 4376, 4384, 4576, 4586 \l__enumext_store_active_bool 30, 67, <u>112</u>, 1911, 1920, 1988, 2620, 3532, 3545, 3693, 3701, 4053, 4578, 4588, 4811, 4827 __enumext_store_active_keys:n 72, 73, 98, 2227, 2227, 3525 __enumext_store_active_keys_vii:n 72,73,117, 2227, 2237, 4557 __enumext_store_addto_prop:n 74, 86, 2302, 2302, 2310, 2470, 2930, 4927 __enumext_store_addto_seq:n 74, 87, 2311, 2311, 2315, 2322, 2336, 2344, 2353, 2367, 2375, 2528, 3023 \l__enumext_store_anskey_arg_tl 30, 77, 78, 112, 2476, 2481, 2483, 2488, 2495, 2498, 2508, 2513, 2516,

- 2821, 2883 \l__enumext_store_columns_break_bool . 2478, 2569 \l__enumext_store_current_label_tl 30, 86, 87, 125, 112, 2911, 2914, 2917, 2923, 2928, 2930, 2987, 2990, 2993, 2999, 3004, 3014, 3023, 4904, 4909, 4913, 4926, 4927, 4929 112, 3319, 3324 \l__enumext_store_current_opt_arg_tl 30, 125, <u>112</u>, 3033, 3046, 3052, 4915 __enumext_store_internal_ref: .. 76, 77, 2392, 2392, 2473 $\g_{\text{enumext_store_item_join_int}}$.. 2723, 2828, 2832, 2884 \l__enumext_store_item_join_int .. 2486, 2490, 2572 \g__enumext_store_item_star_bool . 2725, 2835, 2885 \l__enumext_store_item_star_bool . 2493, 2574 \g__enumext_store_item_symbol_sep_dim 2730, 2850, 2855, 2887 \l__enumext_store_item_symbol_sep_dim 2505, 2510, 2579 \g__enumext_store_item_symbol_tl . 2728, 2841, 2845, 2886 \l__enumext_store_item_symbol_tl . 2496, 2500, \l__enumext_store_keyans_item_opt_sep_tl 2180, 2921, 2925, 2997, 3001, 4907, 4911 __enumext_store_level_close: . 74, 2316, 2340, __enumext_store_level_close_vii: . 75, 2347, 2371, 4592 __enumext_store_level_open: 74, 99, 2316, 2316, 3540, 3553 __enumext_store_level_open_vii: .. 75, 2347, \g__enumext_store_name_tl 30, 67, 112, 363, 370, 371, 372, 373, 1964, 1990, 2113, 2118, 2123, 2137, 2142, 2147, 2893 \l__enumext_store_name_tl 30, 67, 68, <u>112</u>, 1797, 1800, 1824, 1827, 1915, 1924, 1959, 1968, 1969, 1990, $1991,\,1992,\,1994,\,1995,\,1997,\,1999,\,2000,\,2002,\,2004,$ 2005, 2029, 2304, 2306, 2313, 2456, 2457, 2559, 2802, 2970, 2971, 3084, 3097, 4944 \l__enumext_store_ref_key_bool 77, 2203, 2471, 2519, 2934, 3011 \l__enumext_store_save_key_vii_bool . . 2239, \l__enumext_store_save_key_vii_tl 2241, 2242, 2270, 2271, 2351, 2359, 2363, 2367 \l__enumext_store_save_key_X_bool .. 72, 130 \l__enumext_store_save_key_X_tl .. 72, 73, 130 \l__enumext_store_upper_level_X_bool .. 130 __enumext_storing_exec: 67, 82, 1966, 1982, 1986 __enumext_storing_set:n 66, 67, 1951, 1966, 1966 \l__enumext_the_counter_v_tl 741 \l__enumext_the_counter_vii_tl 673 \l__enumext_the_counter_viii_tl 689 \l__enumext_the_counter_X_tl 50 __enumext_tmp:n 45, 49, 54, 60, 71, 78, 79, 86, 94, 99, 100, 111, 134, 141, 166, 170, 174, 194, 514, 523, 1704, 1715, 1947, 1955, 2008, 2026, 2167, 2208, 2209, 2226, 2245, 2258, 2394, 2401, 2402, 2423, 2436, 2439, 2450,
- 2936, 2943, 3243, 3250, 3283, 3290, 3423, 3463, 3464, __enumext_tmp:nn 524, 545, 546, 580, 581, 596, 785, $810,\,887,\,909,\,910,\,930,\,984,\,992,\,993,\,1007,\,1072,$ 1088, 1089, 1102, 1593, 1609, 3227, 3242 __enumext_tmp:nnn 597, 613, 614, 615, 616, 643, 659, __enumext_tmp:nnnnnn 811, 836, 839, 842, 844, 846, 849, 852 __enumext_tmp:w 5061, 5064 \l__enumext_tmpa_vii_int 4202, 4205, 4214, 4245 \l__enumext_tmpa_viii_int 4233, 4236 \l__enumext_tmpa_X_dim 174 \l__enumext_tmpa_X_int 174 \l__enumext_topsep_v_skip 1169, 1173, 1372, 4046 \l__enumext_topsep_vii_skip . . 1449, 1458, 1462 \l__enumext_topsep_viii_skip . 1471, 1493, 1497 __enumext_undefine_anskey_env: . 81, 85, 2653, 2653, 2904 __enumext_unskip_unkern: .. 33, 238, 238, 1222, 1394, 3602, 3603, 3643, 3763, 3764, 3781, 4735, 4736, 4999, 5000 \l__enumext_vspace_a_star_v_bool 1642 \l__enumext_vspace_a_star_vii_bool ... 1664 \l__enumext_vspace_a_star_viii_bool . . . 1675 \l__enumext_vspace_a_star_X_bool 100 __enumext_vspace_above: 60, 100, 1610, 1610, 3613 __enumext_vspace_above_v: . 60, 1638, 1638, 3717 \l__enumext_vspace_above_v_skip . . 1640, 1644, __enumext_vspace_above_vii: 61, 118, 1660, 1660, \l__enumext_vspace_above_vii_skip 1662, 1666, __enumext_vspace_above_viii: . 61, 1660, 1671, \l__enumext_vspace_above_viii_skip 1673, 1677, 1679 \l__enumext_vspace_b_star_v_bool 1653 \l__enumext_vspace_b_star_vii_bool ... 1686 \l__enumext_vspace_b_star_viii_bool . . . 1697 \l__enumext_vspace_b_star_X_bool <u>100</u> __enumext_vspace_below: 60, 101, 1624, 1624, 3651 __enumext_vspace_below_v: . 60, 1649, 1649, 3790 \l__enumext_vspace_below_v_skip . . 1651, 1655, __enumext_vspace_below_vii: 61, 118, 1682, 1682, \l__enumext_vspace_below_vii_skip 1684, 1688, __enumext_vspace_below_viii: . 61, 1682, 1693, 4850 \l__enumext_vspace_below_viii_skip 1695, 1699, __enumext_widest_from:nNNn . . 44, 769, 769, 784, \g__enumext_widest_label_tl 28, 38, 67, 502, 506, \l__enumext_wrap_label_opt_v_bool 3313 \l__enumext_wrap_label_opt_vii_bool 119, 4629 \l__enumext_wrap_label_opt_viii_bool .. 125, \l__enumext_wrap_label_opt_X_bool 100 \l__enumext_wrap_label_v_bool 3309, 3313, 3321, 3368, 3383

\lenumext_wrap_label_vii_bool 119, 4629, 4633, 4641, 4705	2490, 2500, 2510, 2524, 2525, 2832, 2845, 2855, 2980, 3018, 3020, 3930, 5166, 5176, 5369, 5374
\lenumext_wrap_label_viii_bool . 125, 4885,	3010, 3020, 3930, 3100, 3170, 3309, 3374
4889, 4897, 4969	F
\lenumext_wrap_label_X_bool 100	- \fbox 2174
\enumext_wrapper_label_v:n . 3370, 3385, 4109	\fboxrule 2174
\enumext_wrapper_label_vii:n 4707	\fboxsep 2174
\enumext_wrapper_label_viii:n 4971	file commands:
\lenumext_write_aux_file_tl . 31, 77, 87, 163,	\file_input_stop:
2459, 2465, 2977, 2983	first 993
enumext* 5, 4509	font
enumXi 474	\footnote 116
enumXii 474	\footnote 116, 4484
enumXiii 474	\footnotemark 4494
enumXiv 474	\footnotesize 2557, 3082, 3095, 4942
enumXv	\footnotetext 4478
enumXvi	\foreachkeyans 17, 132, 5302
enumXvii <u>474</u>	G
enumXviii <u>474</u>	\getkeyans 17, 127, <u>5050</u>
Environments provide by enumext:	group commands:
anskey* 30, 67, 73, 76, 78, 81–83, 85, 98, 99, 118, 127, 129,	\group_begin: 2555, 2600, 2775, 2862, 3080, 3093,
134, 136	4940, 5107
enumext* 27, 28, 31–33, 36, 38, 39, 41, 42, 44, 46, 48–50,	\group_end: 2562, 2616, 2879, 3087, 3100, 4947, 5114
56, 57, 61–64, 66–69, 71–77, 79, 81, 84–86, 92, 97–99, 104, 111, 112, 114–116, 118, 120–124, 126–131, 135,	
104, 111, 112, 114–110, 110, 120–124, 120–131, 135, 138, 139	Н
enumext 27, 28, 32, 33, 36, 38–46, 48–52, 55, 58–64, 66–69,	\hbadness 4746, 5014
71–74, 76, 77, 79, 81, 84–86, 89–93, 95, 96, 99, 101, 102,	hbox commands:
106, 111, 114, 117, 118, 120, 123, 128–131, 135, 136, 138	\hbox_overlap_left:n 3161, 4698
keyans* 27, 28, 30–34, 38, 41–44, 46, 48–50, 56, 57, 61, 67,	\hbox_set:Nn 494, 3976
68, 71, 72, 74, 81, 86, 92, 97, 104, 112, 113, 116, 122,	\hbox_set_end:
123, 135, 137, 139	\hbox_set_to_wd:\nw 4716, 4980
keyanspic 27, 28, 30, 31, 34, 38, 39, 42, 67, 68, 71, 74, 81,	\hfill 554, 559, 565, 566, 1549, 1576, 2524, 3016, 4379, 4443 hook commands:
86–88, 92, 104–110, 137	\hook_gput_code:nnn 9, 204, 208, 212, 409
keyans 27, 28, 30, 31, 33, 34, 38, 39, 42, 44, 46, 48, 49, 52,	\hook_gput_code:\n\n\ 9, 204, 206, 212, 409 \hook_gremove_code:\n\ 84, 2791
55, 58–60, 67, 68, 71, 72, 74, 81, 86–88, 92–96, 102, 104,	\hook_gset_rule:nnnn 410
106, 107, 110, 114, 124, 135, 137	\hook_if_empty:nTF
Environments:	\hyperlink
center 111	\hyperlink 2524, 3016
description 111	\hypertarget
enumerate 111	\hypertarget 441
flushleft 111	
flushright 111	I
itemize 111	\IfDocumentMetadataTF 3178, 3348, 3847, 3855, 3863, 3897,
list . 32, 35, 36, 79, 95, 100, 101, 104, 106, 107, 111, 114	3905, 3913, 3999, 4009, 4017, 4023, 4028, 4074, 4083,
lrbox 121	4167, 4175, 4377, 4441, 4520, 4528, 4674, 4722, 4748,
minipage 32, 35, 36, 50, 52, 53, 106, 108, 110, 111, 114, 121	4795, 4803, 4986, 5016
multicols 50–53, 58, 99, 100	\IfHyperBoolean 417
quotation 111	\IfPackageLoadedTF
quote	\ignorespaces . 943, 956, 968, 980, 4525, 4602, 4635, 4658,
scontents	4665, 4711, 4737, 4800, 4858, 4891, 4922, 4975, 5001
tabbing 111	\inputlineno 294, 309, 322, 330, 338
trivlist	int commands:
verbatim	\int_add:Nn 4287, 4336
Verse	\int_case:nn 1117, 1242, 2039, 2065, 2104, 2128
exp commands:	\int_case:nnTF
\exp_after:wN	\int_compare:nNnTF 395, 664, 680, 699, 706, 1212,
\exp_args:Ne 2865, 2873, 3522, 5052 \exp_args:NV 2585, 2740, 3253, 3271, 3293, 5337	1231, 1385, 1403, 1515, 1531, 1543, 1571, 2152, 2158,
\exp_args:Nv 2505, 2740, 3253, 3271, 3293, 5337 \exp_not:N . 58, 505, 632, 676, 692, 744, 940, 943, 954,	2624, 2628, 2632, 2640, 2686, 2690, 2694, 2891, 2912,
955, 956, 967, 968, 979, 980, 2524, 2556, 2557, 3016,	2953, 2958, 2963, 2988, 3076, 3504, 3515, 3537, 3550, 3566, 3581, 3596, 3637, 3702, 3706, 3734, 3759, 3775,
3081, 3082, 3094, 3095, 4941, 4942, 5061	3500, 3501, 3590, 3037, 3702, 3700, 3734, 3759, 3775, 3948, 4057, 4061, 4257, 4267, 4283, 4306, 4316, 4332,
\exp_not:n 294, 309, 322, 330, 338, 571, 591, 632, 676,	4538, 4542, 4580, 4590, 4755, 4767, 4816, 4828, 5023,
692, 744, 941, 1731, 1740, 2191, 2288, 2300, 2462,	5035, 5220, 5352
-/-,/11,/1-, -/1-, -/1-, -/1-, -/1-, -/1-, -/1-, -/1-, -/1-, -/1-,	J-JJ; J; JJJ"

©2024 by Pablo González L

\int_compare_p:nNn 263, 273, 285, 286, 300, 301,	1tem-pos* 78, 79, 82−84
1521, 1522, 2045, 2071, 2407, 2417, 2429, 2430, 2445,	item-star
2486, 2663, 2664, 2675, 2676, 2828, 3547	item-sym* 78, 79, 82-84
\int_decr:N 4286, 4335	Keys for environments provide by enumext:
\int_eval:n 380, 798, 2306, 2457, 2557, 2971, 3082,	above*
3095, 3438, 3483, 4275, 4324, 4942	above
\int_from_alph:n 763,777	after 48, 101, 118, 124
\int_from_roman:n	align
\int_gadd:\Nn 4288, 4337	base-fix
\int_gdecr:N 2048, 2053, 2057, 2061, 2074	before* 48, 100, 118, 123
\int_gincr:N 1881, 1886, 2469, 3026, 3115, 3149, 3328,	before 48
3626, 3726, 4098, 4608, 4684, 4864, 4931	below* 29, 59-61, 101, 118
\int_gset:Nn 2097, 4492	below
\int_gset_eq:NN 1780, 1787, 1793, 1799, 1807, 1814,	check-ans . 31-33, 66-71, 74, 85, 88, 101, 118, 122, 136
1820, 1826, 4489	columns-sep
\int_gzero:N . 351, 352, 353, 1557, 1584, 2164, 2884,	columns
3642, 3780, 4778, 5047	first 48, 121
\int_if_exist:NTF 1755, 1791, 1797, 1818, 1824, 2002	font 39, 91, 94, 109, 120
\int_incr:N 2639, 3503, 3697, 3947, 4537, 4607, 4815,	item-pos*
4863	
\int_mod:nn 4769, 5037	item-sym*
	itemindent 29, 46, 47, 89, 90, 93, 94, 121
\int_new:N . 28, 29, 30, 31, 32, 33, 61, 62, 87, 104, 123,	itemsep 44, 97, 122
144, 145, 156, 157, 158, 160, 171, 177, 178, 179, 180,	label-pos
181, 1757, 2005	label-sep 106
\int_set:Nn 759, 763, 765, 1894, 1901, 1913, 1922, 2776,	labelsep 39, 96, 120
4161, 4162, 4202, 4233, 4256, 4262, 4278, 4305, 4311,	labelwidth
4327, 4746, 5014, 5216, 5354	label 28, 38, 40, 43, 44, 107, 111
\int_set_eq:NN 1882, 1887, 4285, 4334	layout-sep 106
\int_sign:n 2099	layout-sty 106, 108, 110, 111
\int_step_function:nnN 2423, 2436, 2450	layout-top 106
\int_step_function:nnnN 5358	lisparindent 97
\int_step_inline:nn 5268	
\int_step_inline:nnn 4163	list-indent 29, 46, 107
\int_to_roman:n	list-offset
	listparindent 46, 121
\int_use:N 373, 378, 379, 1213, 1232, 1544, 1896, 1903,	mark-ans 71, 74, 78
1915, 1924, 3438, 3458, 3483, 3523, 3567, 3576, 3591,	mark-pos 71, 72, 134
3597, 4260, 4261, 4273, 4309, 4310, 4322, 5689, 5693,	mark-ref 71, 74, 76, 78
5699, 5703	mini-env 29, 36, 49, 58, 59, 74, 100, 111, 114, 115, 118, 123
\int_zero:N 4759, 5027	mini-right* 29, 32, 50, 74, 114, 115, 118
\item . 89, 93, 118, 121, 124, 126, 386, 2324, 2330, 2355, 2361,	mini-right 29, 32, 50, 57, 74, 114, 115, 118
2483, 2990, 2993, 3167, 3332, 4003, 4005, 4522, 4524,	mini-sep
4797, 4799, 4929	mode-box
\item* 5, 15, 71, 3330	no-store
item-pos* 3227	
item-sym* 3227	noitemsep
\itemindent	nosep
•	parindent 97
\itemindent	parsep 44, 97, 107, 121
itemindent $\underline{887}$	partopsep 44
\itemsep 3992	ref 28, 32, 40-42, 135
\itemwidth . 464, 2174, 3657, 3663, 3794, 3800, 4296, 4300,	resume* 28, 61, 62, 66, 67, 73, 101, 118, 130
4345, 4349	resume 28, 35, 61-67, 73, 74, 101, 118, 130
	rightmargin
K	save-ans 30, 35, 62-68, 70, 72-74, 79-82, 85-87, 93, 102,
keyans	108, 109, 120, 123–125, 127, 128, 130, 135
keyans*	save-key
keyanspic	
	save-pos
Keys for \anskey provide by enumext:	save-ref 31, 37, 71, 74, 76–78, 86, 87, 94, 125
break-col	save-sep
item-join	series 28, 61-66, 74, 98, 101, 117, 118, 130
item-pos*	show-ans 71, 72, 74, 75, 77, 78, 94, 109, 125
item-star	show-length
item-sym* 78, 79, 82-84	show-pos 30, 71, 72, 75, 77, 78, 88, 94, 109, 125
Keys for anskey* provide by enumext:	start* 29, 43, 44, 62
break-col	start 29, 32, 43, 44, 62
item-join	store-key 72
*****	•

topsep 44, 46, 107	\makelabel 89, 91, 94, 111
widest 28, 32, 44	\makelabel 89, 93, 3194, 3210, 3364, 3378
wrap-ans 37, 71, 74, 75, 78	\makesavenoteenv 435
wrap-label* 29, 39, 89, 91, 93, 94, 119, 120, 125	mark-ans 2167
wrap-label 29, 39, 89-91, 93, 94, 107, 109, 119, 120, 125	mark-pos
wrap-opt 71, 74, 94, 109	mark-ref 2167
keys commands:	mini-env 1072
\keys_define:nn 516, 526, 548, 583, 599, 645, 714, 787,	mini-sep 1072
813, 855, 889, 912, 986, 995, 1074, 1091, 1595, 1706,	\minipage 390
1949, 2010, 2169, 2211, 2247, 2252, 2567, 2718, 2754,	\miniright 11, 58, 1513, 1561, 1588, 3640, 3778
3229, 3245, 3265, 3285, 3919, 5078, 5178, 5294, 5302	mode commands:
\keys_if_exist_p:nn 5290, 5291	\mode_if_math:TF 2648, 2702
\l_keys_key_str 79, 82, 2585, 2740, 3253, 3271, 3293,	\mode_if_vertical:TF 1151, 1179, 1199, 1223, 1374,
5337,5445 \keys_precompile:nnN <i>129</i> ,200,200,5080,5084,	1395
5088, 5092, 5096, 5100, 5320	\mode_leave_vertical: 870, 877, 940, 954, 2380,
\keys_set:nn . 540, 881, 1097, 1600, 1605, 1843, 1848,	3159, 4696
1935, 1943, 2605, 3517, 3522, 3713, 3937, 3940, 3958,	$mode-box\ \dots \ \underline{514}$
4555, 4837, 5182, 5187, 5188, 5189, 5190, 5193, 5198,	msg commands:
5199, 5200, 5201, 5202, 5203, 5204, 5236, 5346	\msg_error:nn 1563, 1590, 2609, 2642, 2646, 2700,
\keys_set_known:nn 2872	2808, 3704, 3708, 3950, 4007, 4059, 4540, 4818, 4830,
keyval commands:	5205, 5264
\keyval_parse:NNn 1720, 2277, 5154	\msg_error:nnn 622, 668, 684, 736, 1517, 1524, 1529,
	1559, 1586, 1855, 1859, 1974, 2591, 2650, 2668, 2680,
L	2688, 2692, 2696, 2704, 2746, 3259, 3277, 3299, 4544,
label	4823, 5066, 5075, 5147, 5252, 5283, 5292, 5329, 5350 \msg_error:nnnn 2594, 2622, 2626, 2630, 2634, 2749,
label-pos <u>3919</u>	3262, 3280, 3302, 3695, 4055, 4063, 4813, 5126, 5332
label-sep	\msg_error:nnnnn 570, 590, 2190, 3929
Labels provide by enumext:	\msg_fatal:nn
\Alph* 38	\msg_fatal:nnn 468
\Roman* 38	\msg_info:nnn
\alph*	\msg_line_context: 5410, 5415, 5420, 5449, 5454,
\arabic*	5459, 5474, 5489, 5493, 5497, 5501, 5505, 5509, 5516,
\labelsep	5523, 5529, 5543, 5547, 5552, 5556, 5560, 5564, 5569,
labelsep 524	5573, 5577, 5581, 5586, 5621, 5625, 5630, 5635, 5639,
\labelwidth	5644, 5720, 5724, 5729, 5734, 5739, 5743, 5747, 5751,
\labelwidth	5755, 5759, 5763, 5767, 5771
labelwidth 524	\msg_log:nnn 1994, 1999, 2004
\lastkern 249	\msg_log:nnnnn 377, 2137, 2142, 2147
\lastnodetype	\msg_log:nnnnnn 369 \msg_new:nnn 5377, 5381, 5385, 5389, 5394, 5407, 5412,
\lastskip 244	5417, 5422, 5431, 5439, 5443, 5447, 5452, 5457, 5472,
layout-sep	5487, 5491, 5495, 5499, 5503, 5507, 5511, 5520, 5526,
layout-sty	5532, 5536, 5540, 5545, 5550, 5554, 5558, 5562, 5567,
layout-top	5571, 5575, 5579, 5584, 5619, 5623, 5628, 5633, 5637,
\leftmargin 96	5642, 5718, 5722, 5727, 5732, 5737, 5741, 5745, 5749,
\leftmargin 95, 3988	5753, 5757, 5761, 5765, 5769
legacy commands:	\msg_new:nnnn 5398, 5589, 5598, 5607, 5613, 5646,
\legacy_if:nTF 4669, 4672, 4954, 4957	5656, 5666, 5676, 5686, 5696, 5706, 5712
\legacy_if_gset_false:n 400, 4392	\msg_term:nnnn . 1958, 1963, 3447, 3457, 3489, 3494
\legacy_if_set_false:n 4671, 4956	\msg_term:nnnnn
\legacy_if_set_true:n 4634, 4657, 4664, 4678, 4890,	\msg_warning:nn 3639, 3777
4921 \linewidth 100	\msg_warning:nnnn 2155, 2161, 3395, 3400, 4259, 4272,
\linewidth 100 \linewidth 3621, 3657, 3723, 3794, 4160, 4205, 4236, 4358,	4308, 4321
4423	\msg_warning:nnnnn
\list 384	\multicolsep
list-indent 887	,
list-offset	N
\listparindent	\NeedsTeXFormat 3
listparindent	\NewCommandCopy 386
	\newcounter 471
M	\NewDocumentCommand 1513, 2597, 4051, 5050, 5105, 5212,
\makebox 111	5261, 5339
\makehov 2282 2284 2212 2280 4001 4702 4066	\NewDocumentEnvironment 2660 2802 2004 4500 4785

\newenvsc 2711	\ref 76, 86
\newlabel 37	ref 597, 643, 714
\newlabel 453	\refstepcounter $\dots \dots \dots$
no-store <u>2008</u>	regex commands:
\noindent 3628, 4367, 4432, 4758, 5026	\regex_match:nnTF 220, 762, 764, 776, 778, 2804
\nointerlineskip 1225, 1228, 1397, 1400, 1551, 1578, 4367,	\regex_replace_once:nnN 228
4432	\renewcommand 632, 676, 692, 744
noitemsep	\RenewDocumentCommand 1561, 1588, 3167, 3194, 3210, 3332,
\nopagebreak 1162, 1190, 1225, 1228, 1397, 1400, 1504, 1510	3364, 3378, 4005, 4484
\normalfont 2556, 3081, 3094, 4941	\RequirePackage 17, 25
nosep	$\begin{tabular}{lllllllllllllllllllllllllllllllllll$
P	resume*
Packages:	rightmargin
caption 114	\Roman
enumext 27, 37, 40, 66, 95, 96, 106, 134	\Roman 490
enumitem 38	\roman 38, 43, 44
expl3	\roman 491, 615, 5095
footnotehyper	S
hyperref	\s 2805
ltcmd	save-ans
lua-visual-debug	
multicol 27, 134	save-key
scontents	save-ref <u>2167</u>
shortlst 111, 116, 121	save-sep
\par 1162, 1190, 1228, 1400, 1504, 1510, 1546, 1551, 1573,	scan commands:
1578, 2532, 3604, 3765, 3783, 4037, 4040, 4180, 4394,	\scan_stop: 4003, 4522, 4797, 5061, 5064
4409, 4455, 4469, 4758, 5026	scontents internal commands:
para commands:	\l_scontents_fname_out_tl 2764
\para_end: 4775, 5044	\scontents_parse_environment_keys:n . 2770
\parbox 2174	\scontents_rescan_tokens:n 2777
\parindent 4733, 4997	\l_scontents_storing_bool 2762
\parsep 51, 107	\lscontents_writing_bool 2763
\parsep 871, 3480, 3972, 3981	seq commands:
parsep	\seq_clear:N 5214, 5357
\parskip 4734, 4998	\seq_const_from_clist:Nn5207
\partopsep 3481, 3781, 3991	\seq_count:N
partopsep	\seq_gclear:N 4482, 4483
peek commands:	\seq_gput_right:Nn 2313, 4495, 4496
\peek_meaning:NTF 4613, 4627, 4642, 4653, 4869, 4883,	\seq_if_empty:NTF 4501, 5120, 5232
4898	\seq_if_exist:NTF
\peek_meaning_remove:NTF 4620, 4876	\seq_if_in:NnTF 5124
\peek_remove_spaces:n 3336	\seq_item:Nn 2802, 4173
\phantomsection 37	\seq_map_function:NN
\phantomsection 442	\seq_map_inline:\Nn 5133, 5141, 5233, 5234
prg commands:	\seq_map_pairwise_function:NNN 4503
\prg_do_nothing: 446	\seq_new:N 124, 125, 127, 142, 172, 173, 2000
\prg_new_protected_conditional:Npnn 218	\seq_pop_left:NN
\prg_replicate:nn 235	\seq_put_right:Nn 4065, 5230, 5246, 5367
\prg_return_false: 222	\seq_set_from_clist:Nn
\prg_return_true: 221	\seq_set_map_e:NNn 5224
\printkeyans	\seq_use: Nn 200, 201, 5363
prop commands:	series
\prop_const_from_keyval:Nn 5253	\setcounter 773, 777, 779, 3438, 3483, 4034
\prop_count:N 371, 2306, 2457, 2559, 2971, 3084, 3097,	\setenumext 6, 130, <u>5212</u>
4944, 5355	\setenumextmeta 6, 131, <u>5253</u>
\prop_get:NnNTF 5279	show-ans
\prop_gput_if_not_in:Nnn 2304	show-length <u>984</u>
\prop_if_exist:NTF 1992, 5070, 5348	show-pos
\prop_item:\Nn 5072, 5372	skip commands:
\prop_new:N	\skip_add:Nn 1122, 1131, 1140, 1153, 1157, 1181, 1185,
\ProvidesExplPackage 4	1201, 1259, 1261, 1275, 1278, 1299, 1301, 1315, 1318,
	1338, 1340, 1354, 1357, 1376, 1425, 1426, 1437, 1439,
R	3981, 3989
\raggedcolumns 3590, 3753	\skip_gset:Nn 1452, 1456, 1460
\raisebox 4122	\skip_gzero_new:N 1447, 1448

\skip_horizontal:N 955, 967, 979, 4699, 4711, 4762,
4975, 5030 \skip_horizontal:n 941, 2381, 2389, 3160, 3162,
4598, 4697, 4737, 4854, 5001
\skip_if_eq:nnTF 1120, 1129, 1138, 1245, 1285, 1325, 1413, 1449, 1471, 1612, 1626, 1640, 1651, 1662, 1673,
1684, 1695
\skip_new:N 81, 82, 83, 88, 89, 90, 91, 92, 93, 148, 192
\skip_set:Nn 1105, 1109, 1167, 1171, 1195, 1248, 1249,
1267, 1288, 1289, 1307, 1327, 1328, 1346, 1370, 1416,
1417, 1431, 1451, 1455, 1473, 1477, 1481, 1487, 1491,
1495, 3965
\skip_set_eq:NN 1206, 1207, 1209, 1216, 1381, 1382,
1383, 1388, 3436, 3479, 3480, 4734, 4998
\skip_sub:Nn 1255, 1257, 1271, 1273, 1295, 1297, 1311,
1313, 1334, 1336, 1350, 1352, 1423, 1424, 1435, 1436
\skip_use:N 1107, 1111, 1155, 1159, 1163, 1183, 1187, 1197, 1203, 1613, 1617, 1620, 1627, 1631, 1634, 3604
\skip_vertical:N . 401, 404, 879, 4393, 4407, 4777,
5046
\skip_vertical:n 878, 4776, 5045
\skip_zero:N 1215, 1229, 1367, 1368, 1369, 1387, 1401,
3481, 3587, 3750, 3991, 3992
\skip_zero_new:N 1446, 1468, 1469, 1470
\c_zero_skip . 401, 404, 879, 1120, 1129, 1138, 1286,
1325, 1449, 1471, 1613, 1627, 1640, 1651, 1662, 1673,
1684, 1695, 4393, 4407, 4777, 5046
\small 5083, 5087, 5091, 5095, 5099, 5103
socket commands:
\socket_assign_plug:nn 3849, 3857, 3865, 3899,
3907, 3915
\socket_new:nn 3821, 3869
\socket_new_plug:nnn 3822, 3829, 3837, 3870, 3877,
3886
\socket_use:n 3850, 3900
\socket_use:nn 3858, 3866, 3908, 3916
start <u>785</u>
start* <u>785</u>
start-list-tags <u>3821,</u> <u>3869</u>
\stepcounter
stop-list-tags <u>3821</u> , <u>3869</u>
stop-start-tags <u>3821</u> , <u>3869</u>
str commands:
\c_backslash_str 2650, 5410, 5415, 5420, 5425, 5427,
5429, 5434, 5436, 5534, 5538, 5542, 5552, 5556, 5564,
5565, 5569, 5581, 5582, 5586, 5587, 5608, 5610, 5614,
5616, 5644, 5707, 5709, 5713, 5715, 5724, 5725, 5729, 5734, 5735, 5739, 5743, 5747
\c_colon_str 2456, 2970, 5061
\c_left_brace_str 5515, 5522, 5528
\c_right_brace_str 5515, 5522, 5528
\str_case:nn 256, 315
\str_case:nnTF . 1727, 1735, 2284, 2292, 5161, 5170
\str_clear:N 3514, 4554
\str_count:n 235
\str_if_empty:NTF 1744, 1785, 1812
\str_if_eq:nnTF 3439, 3485, 5263
\str_if_in:nnTF 5057
\str_new:N 84, 132, 147, 187
\str_set:Nn . 555, 561, 567, 586, 587, 588, 2187, 2188,
2214, 2215, 3924, 3927
\str_use:N 3214
\string 435
\strutbox . 1234, 1237, 1248, 1249, 1260, 1262, 1277, 1280,
1288, 1289, 1300, 1302, 1317, 1320, 1327, 1328, 1339,

1341, 1356, 1359, 1405, 1408, 1416, 1417, 1425, 1426, 1438, 1440, 1451, 1452, 1455, 1462, 1475, 1483, 1489, 1497, 3984, 3989, 4037, 4045, 4128

Т

tag commanus.
\tag_mc_begin:n 3827, 3875, 3884
\tag_mc_end: 3831, 3879, 3888
\tag_resume:n 3824, 3872, 4011, 4019, 4085, 4177
4377, 4441
\tag_struct_begin:n . 3825, 3826, 3833, 3834, 3835,
3873, 3874, 3881, 3882, 3883, 4020
\tag_struct_end: 4030, 4031
\tag_struct_end:n 3832, 3839, 3840, 3841, 3842, 3880,
3889, 3890, 3891, 3892, 4528, 4803
\tag_suspend:n . 3843, 3893, 4001, 4013, 4025, 4076,
4169, 4520, 4795
\tag_tool:n 4012
Τ _E X and ⊮Τ _E X 2 _ε commands:
\@auxout 451
\@currenvir 256, 315
\protected@write 451
tex commands:
\tex_newlinechar:D 2776
text commands:
\text_expand:n 5053
\textasteriskcentered 2184, 3233
\textreferencemark
\the
\thepage
tl commands:
\c_space_tl 3052, 5459, 5474, 5497, 5501, 5688, 5689,
5698, 5699, 5759, 5763
\tl_clear:N 553, 560, 2165, 2231, 2241, 2262, 2270,
2476, 2796, 2797, 2911, 2987, 4904
\tl_clear_new:N 500
\tl_const:Nn 50, 484
\tl_gclear:N . 363, 364, 365, 1765, 1770, 2886, 3205,
3223, 4413, 4473, 4700
\tl_gclear_new:N1752
\tl_gput_right:Nn 485
\tl_greplace_all:Nnn 506
\tl_gset:Nn 291, 292, 306, 307, 1753, 1766, 1771, 1990,
2800, 3136, 4648
\tl_gset_eq:NN 502, 3132, 4693
\tl_if_blank:nTF 2589, 2607, 2744, 3257, 3275, 3297,
4691, 5327
\tl_if_empty:NTF . 620, 638, 666, 682, 701, 708, 734
750, 1778, 1783, 1805, 1810, 1868, 1932, 1940, 1969,
2029, 2320, 2351, 2496, 2841, 2863, 2893, 2921, 2997,
3046, 3157, 4184, 4907, 5244
\tl_if_empty:nTF 1833
\tl_if_exist:NTF1838
\tl_if_novalue:nTF 2603, 2919, 2995, 3031, 3111,
3130, 3138, 3307, 3512, 3956, 4486, 4552, 4835, 4905
\tl_map_inline:Nn 226, 503
\tl_new:N 42, 43, 44, 47, 52, 53, 56, 57, 63, 65, 66, 68, 69,
105, 106, 107, 113, 114, 115, 116, 117, 118, 119, 120,
121, 122, 126, 128, 129, 130, 133, 136, 137, 155, 163,
164, 165, 168, 186
\tl_put_left::Ne 2830
\tl_put_left:Nn 2328, 2359, 2481, 2824, 2837, 2843,
2853, 3063, 3103, 4397, 4458, 4926, 4929
\tl_put_right:Nn 501, 630, 674, 690, 742, 2332, 2363,
2410 2420 2433 2448 2454 2450 2483 2488 2405

2498, 2508, 2513, 2516, 2522, 2914, 2917, 2923, 2928,

2955, 2960, 2965, 2968, 2977, 2990, 2993, 2999, 3004,	U
3014, 4909, 4913	\u 229, 2805
\tl_remove_all:Nn5243	\unkern 250
\tl_remove_once:Nn 2398, 2940	unknown
\tl_replace_all:Nnn 505, 5278	\unskip 245
\tl_reverse:N 2397, 2399, 2939, 2941	use commands:
\tl_set:Nn . 58, 260, 270, 319, 320, 327, 328, 335, 336,	\use:N 236, 3201, 3220, 3678
470, 554, 559, 565, 566, 619, 663, 733, 938, 952, 965,	\use:n 1718, 2275, 5059, 5152
977, 1867, 1968, 2232, 2242, 2263, 2271, 2553, 2764,	\use_none:nn 445, 5284
3033, 3078, 3091, 4915, 4938, 5241, 5277, 5347	\usecounter 3437, 3482
\tl_set_eq:NN 511, 625, 628, 671, 673, 687, 689, 739, 741, 2396, 2938, 2951, 3319, 3324, 4103, 4105	v
\tl_to_str:n 1838, 1844, 1849, 5053	\value 1781, 1787, 1794, 1800, 1808, 1814, 1821, 1827
\tl_trim_spaces:n 501, 5230, 5241, 5247, 5263	vbox commands:
\tl_use:N 507, 510, 640, 703, 710, 752, 1010, 1014, 1018,	\vbox_set:Nn 4078
1022, 1026, 1030, 1034, 1038, 1042, 1046, 1050, 1054,	\vbox_set_top:\Nn 4402, 4463
1058, 1062, 1066, 1070, 2386, 2403, 2411, 2422, 2435,	\vspace 871, 1617, 1620, 1631, 1634, 1644, 1646, 1655, 1657,
2440, 2451, 3119, 3125, 3153, 3196, 3197, 3204, 3216,	1666, 1668, 1677, 1679, 1688, 1690, 1699, 1701
3310, 3314, 3322, 3366, 3367, 3373, 3382, 3676, 3809,	W
4108, 4404, 4465, 4704, 4738, 4739, 4968, 5002, 5007,	widest 785
5108, 5109, 5110, 5111, 5112, 5129, 5226, 5345	wrap-ans
token commands:	wrap-label
\token_to_str:N 453	wrap-label*
\topsep 3781, 3989	
topsep	wrap-opt <u>2167</u>
\topskip 1215, 1387	7.
\typeout 244, 249, 419, 423, 434, 435	\z
(-) [1.7, 1.7, 1.7]	

©2024 by Pablo González L