

enumitem-zref

Extended references to items for enumitem package

The `\refitem` command

FC

2011/02/18 – version 1.8

Abstract

`enumitem-zref`, as a companion package for `enumitem`¹, extends the references mechanism for lists of items and allows hyperlinks to any items in lists formatted by `enumitem`. Hyper-references to lists seems not always necessary, but their importance rise much when a document contains lists that can stretch out on several pages.

Primarily intended for `enumerate` environments (*ie.* numbered lists), `enumitem-zref` can also reference any item in `description` or `itemize` environments: restrictions can be set with the `package options`.

A `natural reference scheme` for referencing items anywhere in the document is provided with three modalities (the `commaref` scheme, the `refitem` scheme and the `user` scheme). The references can be typeset differently with the command `\refitem`, its options and eventually arbitrary text.

Additionally, `enumitem-zref` allows to typeset greek-enumerated lists with the `greek` or `greekctr` package options, which use code from packages `alphalph`² and `engrec`³ or `greekctr`⁴.

References to items in external documents compiled with `enumitem-zref` is also possible with pdf \TeX ⁵.

`enumitem-zref` requires and is based on the packages `enumitem` by Javier Bezos and `zref`⁶ by Heiko Oberdiek. It works with an ε - \TeX distribution of L \TeX .

Contents

Summary of the properties for <code>\refitem</code>	2	4.5	Normalisation of references and item labels	11
1 Forewords	3	4.6	Keys <code>name</code> and <code>name*</code> are added to <code>enumitem</code> environments	12
1.1 Numbered lists, not numbered lists and <code>hyperref</code>	3	4.7	Overloading <code>\makelabel</code>	14
1.2 A “natural” reference scheme for lists	3	4.8	Creating references	14
1.2.1 The <code>commaref</code> scheme		4.9	Extracting <code>zref</code> properties	16
1.2.2 The <code>refitem</code> scheme		4.10	Referencing the items: <code>\refitem</code>	16
1.2.3 The <code>user</code> scheme: with the <code>\label</code> command		4.11	Undefined references and duplicate labels management	19
2 User Interface	5	4.12	<code>\zeninfo</code> (If you are lost...)	19
2.1 Naming lists	5	4.13	Create the references At Begin Document for the <code>refitem</code> and the <code>commaref</code> schemes	20
2.2 Referencing the items : <code>\refitem</code>	5	5 References		22
2.3 The package options	6	6 History		22
2.4 If your are lost <code>\zeninfo</code> is your friend	7	[2011/02/18 v1.8]		22
3 Flow chart of expansion: How does it work ?	8	[2010/12/30 v1.75]		22
4 IMPLEMENTATION	9	[2010/12/27 v1.7]		22
4.1 Identification and requirements	9	[2010/12/17 v1.5]		22
4.2 The package options	9	[2010/12/10 v1.2]		22
4.3 Some constants	10	[2010/12/02 v1.1]		22
4.4 Properties for the <code>zref</code> list: <code>zen@list</code>	11	7 Index		23

1. `enumitem`: CTAN:help/Catalogue/entries/enumitem.html
2. `alphalph`: CTAN:help/Catalogue/entries/alphalph.html
3. `engrec`: CTAN:help/Catalogue/entries/engrec.html
4. `greekctr`: CTAN:help/Catalogue/entries/greekctr.html
5. This is not `xr` but “pure” PDF.
6. `zref`: CTAN:help/Catalogue/entries/zref.html

This documentation is produced with the DocStrip utility.

- To get the package, run: `etex enumitem-zref.dtx`
- To get the documentation run (thrice): `pdflatex enumitem-zref.dtx`
- To get the index, run: `makeindex -s gind.ist enumitem-zref.idx`

The `.dtx` file is embedded into this pdf file thank to `embedfile` by H. Oberdiek.

Summary of the properties for `\refitem`

Terms definitions

- `\label {user}` This is a user *label*.
- `\item [user]` This is a user *item*.
- `\item` This is an *automatic* item.



User items are labeled, hence expanded (with `\protect`).

The `\label` command can be put inside the optional argument of `\item`:

```
\item [ text \label {label of the item}]
```

or in the paragraph that follows.

Package `hyperref`'s `[verbose]` option reports all inserted anchor names in the `.log` file.

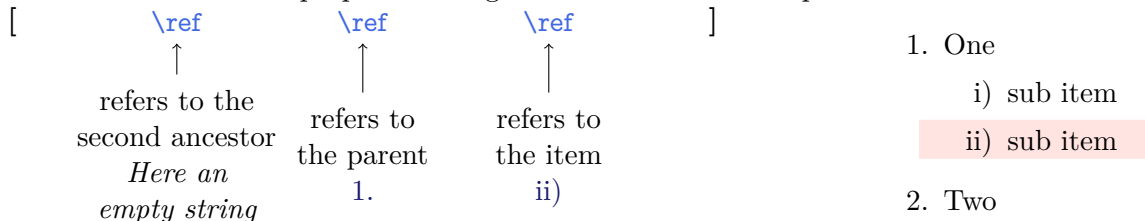
Syntax for `\refitem`

```
\refitem {reference} [properties to display]
```

Optional text containing properties
Default is `[\ref]`

reference with either: $\left\{ \begin{array}{l} \text{The } \textit{refitem} \text{ scheme: } \text{listname}\{\text{label}\}\{\text{label}\}\{\text{label}\}\dots \\ \text{The } \textit{commaref} \text{ scheme: } \text{listname,index,index,index},\dots \\ \text{The } \textit{user} \text{ scheme: } \text{reference to a } \backslash\text{label} \text{ command} \end{array} \right.$

The order in which the properties are given is related to the depth of the list:



	Property	Description	Examples
most common properties	<code>\label</code>	The item label: automatic or user defined by <code>\item [...]</code>	II) or •
	<code>\label*</code>	For automatic labels in <code>enumerate</code> lists: the label without any text around (useful to “concatenate” labels). Otherwise, the same as <code>\label</code> .	II) or •
	<code>\ref</code>	If the <code>ref</code> key of <code>enumitem</code> has been set, the “evaluation” of the <code>ref</code> key. Otherwise, the same as <code>\label</code>	II) or •
	<code>\ref*</code>	The same as <code>\label*</code>	II) or •
	<code>\listname</code>	The name given to the list with the key <code>name</code> .	
	<code>\depth</code>	The list depth.	
	<code>\index</code>	The index of the item in the list. Every item is counted.	
	<code>\listctr</code>	The value of <code>\@listctr</code> for the item – only inside <code>enumerate</code> lists. May be different from <code>\index</code> if the list contains user <i>items</i> .	
	<code>\type</code>	The type of the list: either <code>enumerate</code> or <code>itemize</code> or <code>description</code> .	
	<code>\currentlabel</code>	L ^A T _E X's <code>\@currentlabel</code>	
hyperref	<code>\anchor</code>	The name of the anchor set for the item.	
hyperref	<code>\currentlabelname</code>	<code>hyperref</code> 's (or <code>nameref</code> 's) <code>\@currentlabelname</code>	
	<code>\refitem</code>	The list of references with the <i>refitem</i> scheme for the item.	<code>\zeninfo</code>
	<code>\commaref</code>	The list of references with the <i>commaref</i> scheme for the item.	<code>\zeninfo</code>
	<code>\default</code>	The same as <code>\ref</code>	

1 Forewords

1.1 Numbered lists, not numbered lists and hyperref

L^AT_EX defines two “classes” of lists of items:

- **enumerate** lists which are *numbered* (*ie.* a counter gives an index to each item in the list),
- **description** and **itemize** lists which are *not numbered* (no counter, thus no index).

When `hyperref` is used in *numbered lists*, each `\item` gives `hyperref` the opportunity to define an anchor, whose name is chosen considering the current value of the *index*.

Different cases have to be considered:

- 1) We are in a *numbered list* (*ie.* `enumerate`):
 - i) if the item is automatic (`\item` without option), `hyperref` puts an anchor named `Item.IDX`.
 - ii) if the item is user defined (`\item [user item]`), `hyperref` does not put an anchor.
- 2) We are not in a *numbered list* (*ie.* `description` or `itemize`):

`hyperref` does not put an anchor.

Moreover, when `hyperref` is loaded, the expansion of the `\label` command leads to the expansion of the item label (with `\protect`) for it is written to the `.aux` file:

- in the case 1 i)
- and also in the case 1 ii) if:
 - the list is a `description` list, and
 - package `nameref` is loaded, and
 - the `\label` command appears inside the `user item` text.

I was guardedly afraid about those rules when I understood them, as you may be...

`enumitem-zref` adds an anchor in the cases 1 ii) and 2). The item label text is always expanded (with `\protect` and `babel` protection).

1.2 A “natural” reference scheme for lists

`enumitem-zref` provides **3 ways** to reference a specific item in a list. The first 2 can be used to reference items in lists of external documents as well.

References are made with the command: `\refitem`, but `\hyperref` can be used as well.

`\subitem` and `\subsubitem` provided in some classes cannot be referenced with `enumitem-zref`.

1.2.1 The *commaref* scheme

Assuming you gave a name to your `enumerate`, `description` or `itemize` list, and let’s say this name is ‘`myList`’, each item is given an index. Any item is indexed, whether it’s printed by the mean of a user item or an an automatic item. Then:

- this is the item `myList,1`
 - this is the item `myList,2`
 - this is the item `myList,2,1`
 - this is the item `myList,2,2`
- user-defined** this is the item `myList,2,3`

Reference to the third item can be made with:

```
\refitem {myList,2,1}           this prints: 1 for it is the first item in the inner list.
\refitem {myList,2,1}[\ref.\ref] prints: 2.1.
\refitem {myList,2,1}[\label \ \label] prints: • –.
```

Nothing simplest, no more comment...

1.2.2 The *refitem* scheme

- A) this is the item `myList{A}`.
- B) this is the item `myList{B}` (as you'd have expected...)
- i) this is the item `myList{B}{i}`.
and is *also* `innerlist{i}` if this nested list has been named 'innerlist'.
 - ii) this is the item `myList{B}{ii}` (or `innerlist{ii}`).
 - α this is the item `myList{B}{ii}{\alpha}` (with the package option `[greek]`).
 - β this is the item `myList{B}{ii}{\beta}` (or `innerlist{ii}{\beta}`).
- C) this is the item `myList{C}` (as you'd have expected...)
- perso** this is the item `myList{perso}`.
- D) this is the item `myList{D}` and is also `myList,5` in the *commaref* scheme.
- this is the item `myList{D}{1}`
 - this is the item `myList{D}{2}`
- inside `itemize` lists, labels are replaced by their arabic index in the *refitem* scheme.

And here is a reference to:

```
\refitem {myList{B}{ii}{\alpha }}  $\alpha$ 
\refitem {myList{B}{ii}{\alpha }}[\label \ \label \ \label] B) ii)  $\alpha$ 
\refitem {myList{B}{ii}{\alpha }}[\label*--\label*--\label] B-ii- $\alpha$ 
```

Now what happens with a description list:

- 1°) this is the item `myList{1}`
- 2°) this is the item `myList{2}`
- First description label** this is the item `myList{2}{first description label}`
- And a second one** this is the item `myList{2}{and a second one}`
- 3°) this is the item `myList{3}`

```
\refitem {myList{2}{first description label}} First description label
\refitem {myList2{2}{first description label}}[\ref*, \label] 2, First description label
\refitem {myList2{2}{first description label}}[\ref \ \ref] 2°) First description label
```

The correspondance between the reference name and the reference that is finally printed is immediate: the *refitem* scheme is almost useful when used with `hyperref`.

The list name `myList` is case sensitive, but the item number `{B}` or `{first description label}` is not. This can be changed with the `CaSeS` package option. Trailing spaces in the item label are always removed from the reference name in the *refitem* scheme.

1.2.3 The *user* scheme: with the `\label` command

- 1 – this is the item `myList,1` and `myList{1}` and as long as I type `\label {Important Item}` it is also named `Important Item`.
- 2 – this is the item `myList,2` and `myList{2}`.
- a) this is the item `myList,2,1` and `myList{2}{a}`.
 - b) this is the item `myList,2,2` and `myList{2}{b}` and as long as I type: `\label {Important subitem}` it is also named `Important subitem`.

Then you get:

```
\refitem {Important subitem} b)
\refitem {Important subitem}[\label \ \label] 2 – b)
\refitem {Important subitem}[\ref* sub point \ref] 2 sub point b)
```

The reference is case sensitive.

2 User Interface

2.1 Naming lists

To use the reference schemes defined by `enumitem-zref`, it is necessary to give lists a name. For that purpose, two new keys: `name` and `name*` have been added to the keys provided by `enumitem`.

The key ‘`name`’ gives a name to the list, while `name*` is a bit more clever and should be used in every circumstances when lists are nested, almost when lists are automatically named by the mean of the paragraph they belong to. Example:

```
\newlist{myList}{enumerate}{3}
\setlist{myList}{label* = \arabic*., name* = \arabic{section}.\arabic{subsection}}
```

In case of nested lists, `name*` will compare the name you gave to the list with the name of its parent list, and ignore the key in case of equality.

This gives:

1. First item
2. Second item
 - 2.1. sub item
 - 2.2. another sub item
 - 2.3. a last sub item \label {for further reference}

And then:

<code>\refitem {2.1,2,3}</code>	2.3.	in the <i>commaref</i> scheme
<code>\refitem {2.1{2}{3}}</code>	2.3.	in the <i>refitem</i> scheme
<code>\refitem {for further reference}</code>	2.3.	in the <i>user</i> scheme

To name your lists automatically it is also possible to use (these are examples):

- `interfaces-hybmsec` with package `hybmsec`:
`name* = \csname the\lastsectionname \endcsname`
- package `refcount`: `name* = \getrefbykeydefault {reference}{anchor}{default}`.

2.2 Referencing the items : `\refitem`

`\refitem {<reference>}[text with properties \ref, \label etc..]babel`
`\refitem *`

`enumitem-zref` provides `\refitem` to reference the items of lists with the following possible syntaxes:

<code>\refitem {refname}[properties]</code>	displays a reference to <code>refname</code> the text displayed depends on the optional <code>properties</code> whose default value is <code>\ref</code> .
.....	
<code>\refitem * {refname}[properties]</code>	can be used when you want a reference but no hyperlink.

If the reference `rename` does not point to an item, `\refitem` behaves like `\ref` (or `\ref*`) and the `[properties]` are ignored. A warning is displayed in the `.log` file.

The properties available with `\refitem` are:

- | | |
|-------------------|---|
| <code>\ref</code> | This is the default property: the same as L ^A T _E X’s <code>\ref</code> command: it expands to the text corresponding to the <code>ref</code> key of the <code>enumitem</code> list environment, which by default is: <ul style="list-style-type: none"> • in <code>enumerate</code> lists: the same as the <code>label</code>. • in <code>itemize</code> lists: the index of the item. • in <code>description</code> lists: the user defined label, or if no label has been given to the <code>\item</code> command, the index of the item. |
|-------------------|---|

<code>\label</code>	The label is printed, whatever value the <code>ref</code> key has been given. For <code>description</code> , the label may be empty.
<code>\label* \ref*</code>	This is a variant of <code>\label</code> : automatic labels in <code>enumerate</code> lists are related to a counter. <code>\ref*</code> or <code>\label*</code> prints the formatted value of the counter, without any text around. For <code>itemize</code> , <code>description</code> or user items inside <code>enumerate</code> lists, this is the same as <code>\label</code> .
<code>\page</code>	The same as <code>\pageref</code> .
<code>\name</code>	The name that has been given to the list.
<code>\depth</code>	The depth of the list (equal to 1 if the list is not nested).
<code>\index</code>	The index of the item; user defined items are counted.
<code>\listctr</code>	The item number inside numbered lists (<i>ie.</i> <code>enumerate</code>); user defined items are not counted: this is the value of the counter <code>\@listctr</code> .
<code>\type</code>	The type of the list (<i>ie.</i> <code>enumerate</code> , <code>itemize</code> or <code>description</code>)
<code>\refitem</code>	The list of references names that can be invoked with <code>\refitem</code> for this specific item in the <i>refitem</i> scheme (see also <code>\zeninfo</code>).
<code>\commaref</code>	The list of references names that can be invoked with <code>\refitem</code> for this specific item in the <i>commaref</i> scheme (see also <code>\zeninfo</code>).

`\refitem [file:...] {<reference>} {<text>}babel`

For reference to external files (to make hyperlinks), `\refitem` can be invoked with the following syntax:

<code>\refitem [file:...] {refname} {text}</code>	Can be used for references to an external .pdf file.
<code>\refitem [] {refname} {text}</code>	The current file is used by default...

Only the *refitem* and the *commaref* schemes can be used. The *user* scheme cannot be used presently for external files. An implementation “à la xr” might be provided in a further release.

2.3 enumitem-zref package options

`enumitem-zref` can be loaded with the following options:

defaults ↓	true	<code>enumerate</code>	disables the references for <code>itemize</code> and <code>description</code> lists: only enumerated lists can be referenced.
	true	<code>itemize</code>	enables/disables the references for <code>itemize</code> lists.
	true	<code>description</code>	enables/disables the references for <code>description</code> lists.
	false	<code>greek</code>	Defines <code>\engrec</code> (lowercase) and <code>\EnGrec</code> (uppercase) to make enumerate lists of items with the greek alphabet as the label (package <code>engrec</code>). <code>\engrec</code> and <code>\EnGrec</code> can be used in place of <code>\arabic</code> (f.ex. <code>label=\engrec *</code>).
	false	<code>greekctr</code>	Defines <code>\greek</code> and <code>\Greek</code> (like <code>greek</code> option but with package <code>greekctr</code> ⁷).
	false	<code>CaSeS</code>	Activate this option to get CaSeS sEnSiTiV references in the <i>refitem</i> scheme. This option applies only to the <i>refitem</i> scheme. List names are always case sensitive. If this option is enabled with <code>hyperref</code> , anchors corresponding to the <i>refitem</i> scheme will be case sensitiv as well. Package <code>hyperref</code> ’s <code>[verbose]</code> option allows to see anchor names in the <code>.log</code> file.

	CaSeS = true	CaSeS = false
	I) First	I) First
	II) Second	II) Second
<code>\refitem {myList{I}}</code>	✓works	✓works
<code>\refitem {myList{i}}</code>	✗doesn’t work	✓works
<code>\refitem {mylist{I}}</code>	✗doesn’t work	✗doesn’t work

List names are always case sensitive.

7. `greekctr`: CTAN:help/Catalogue/entries/greekctr.html

Package option	Types of lists that can be referenced		
	<i>enumerate</i>	<i>itemize</i>	<i>description</i>
[enumerate]	•		
[itemize]	•	•	
[description]	•		•
[enumerate,itemize]	•	•	
[enumerate,description]	•		•
[itemize,description]	•	•	•
[enumerate,itemize,description]	•	•	•
none of the above	•	•	•

2.4 If your are lost `\zeninfo` is your friend

`\zeninfo` [info-key]

In the paragraph following an item, `\zeninfo` reports some informations about the references to the current item:

- depth** The depth of the list.
- index** The index of the item.
- commaref** The references that can be used to reference the item in the *commaref* scheme.
- refitem** The reference that can be used to reference the item in the *refitem* scheme.
- anchor** The anchor set for the item (if hyperref).

With no argument, `\zeninfo` prints all these informations.

Example with the `pifont` package:

```
\newcommand\@Ding[1]{\ding{\number\numexpr#1 + 171}}
\newcommand\Ding[1]{\@Ding{\value{#1}}}
\AddEnumerateCounter \Ding \@Ding {}
```

```
\begin{enumerate}[label=\Ding*,name=dingList]
\item One
\item Two
\item Three            \zeninfo
\end{enumerate}
```

① One

② Two

③ Three `zeninfo:depth=1/ idx=3/ ctr=3/ commaref={dingList,3}/ refitem={dingList{\protect \fontfamily {pzd}\protect \fontencoding {u}\protect \fontseries {m}\protect \fontshape {n}\protect \selectfont \par@update \char 174}}/ anchor=Item.37`

Now:

`\refitem {dingList{\@Ding {3}}}` produces ③

Note that for this purpose:

- `\@Ding` could be named differently (without `@`).
- the *commaref* scheme is easier: `\refitem {dingList,3}` gives ③.

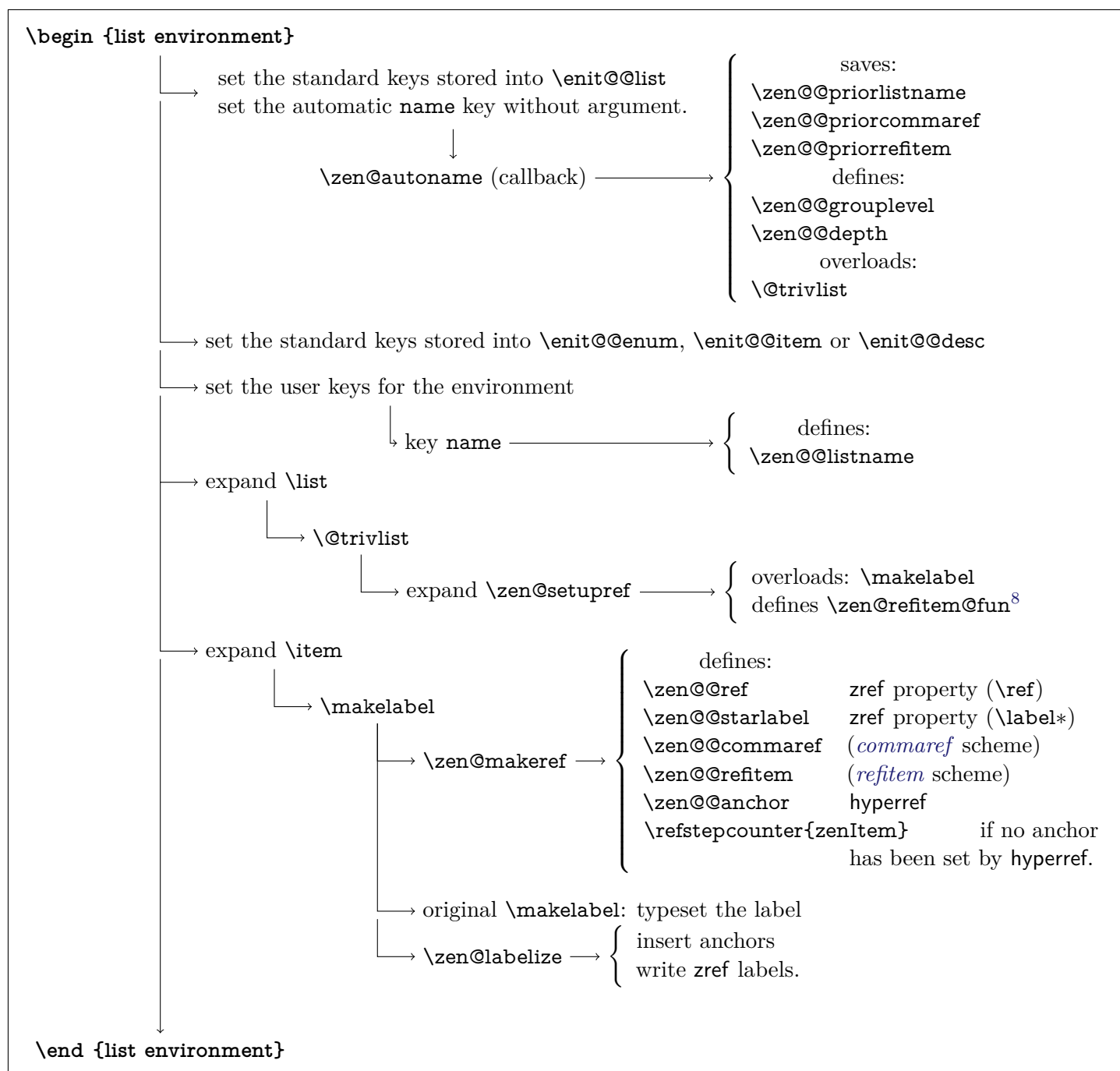
This example shows the output of `\zeninfo`.

7. For the French language with `babel frenchb-setup`, `itemize` lists can be processed by `enumitem-zref` only if the option:

```
\frenchbsetup {CompactItemize=false}
```

has been set. This option is automatically set by `enumitem-zref` when option `[itemize]` is passed.

3 Flow chart of expansion: How does it work ?



8. Used for the property `\label*` and to define the label (and the anchor) in the *refitem* scheme.

4 Implementation

4.1 Identification and requirements

The package namespace is `\zen@`

```

1 (*package)
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{enumitem-zref}[2011/02/18 v1.8 - Extended references for enumitem pkg]
4 \RequirePackage{ltxcmds,etoolbox,kvoptions,getttitlestring,enumitem}
5 \RequirePackage{zref}[2010/05/01]
6 \RequirePackage{zref-user,zref-counter}
7 \GetTitleStringDisableCommands{\let\enit@format \@empty}

```

Some `\catcode` assertions:

```

8 \let\zen@AtEnd \@empty
9 \def\TMP@EnsureCode#1#2#3#4#5#6#7#8#9#10#11#12#13#14#15#16#17#18#19#20#21#22#23#24#25#26#27#28#29#30#31#32#33#34#35#36#37#38#39#40#41#42#43#44#45#46#47#48#49#50#51#52#53#54#55#56#57#58#59#60#61#62#63#64#65#66#67#68#69#70#71#72#73#74#75#76#77#78#79#80#81#82#83#84#85#86#87#88#89#90#91#92#93#94#95#96#97#98#99#100#101#102#103#104#105#106#107#108#109#110#111#112#113#114#115#116#117#118#119#120#121#122#123#124#125#126#127#128#129#130#131#132#133#134#135#136#137#138#139#140#141#142#143#144#145#146#147#148#149#150#151#152#153#154#155#156#157#158#159#160#161#162#163#164#165#166#167#168#169#170#171#172#173#174#175#176#177#178#179#180#181#182#183#184#185#186#187#188#189#190#191#192#193#194#195#196#197#198#199#200#201#202#203#204#205#206#207#208#209#210#211#212#213#214#215#216#217#218#219#220#221#222#223#224#225#226#227#228#229#230#231#232#233#234#235#236#237#238#239#240#241#242#243#244#245#246#247#248#249#250#251#252#253#254#255#256#257#258#259#260#261#262#263#264#265#266#267#268#269#270#271#272#273#274#275#276#277#278#279#280#281#282#283#284#285#286#287#288#289#290#291#292#293#294#295#296#297#298#299#300#301#302#303#304#305#306#307#308#309#310#311#312#313#314#315#316#317#318#319#320#321#322#323#324#325#326#327#328#329#330#331#332#333#334#335#336#337#338#339#340#341#342#343#344#345#346#347#348#349#350#351#352#353#354#355#356#357#358#359#360#361#362#363#364#365#366#367#368#369#370#371#372#373#374#375#376#377#378#379#380#381#382#383#384#385#386#387#388#389#390#391#392#393#394#395#396#397#398#399#400#401#402#403#404#405#406#407#408#409#410#411#412#413#414#415#416#417#418#419#420#421#422#423#424#425#426#427#428#429#430#431#432#433#434#435#436#437#438#439#440#441#442#443#444#445#446#447#448#449#450#451#452#453#454#455#456#457#458#459#460#461#462#463#464#465#466#467#468#469#470#471#472#473#474#475#476#477#478#479#480#481#482#483#484#485#486#487#488#489#490#491#492#493#494#495#496#497#498#499#500#501#502#503#504#505#506#507#508#509#510#511#512#513#514#515#516#517#518#519#520#521#522#523#524#525#526#527#528#529#530#531#532#533#534#535#536#537#538#539#540#541#542#543#544#545#546#547#548#549#550#551#552#553#554#555#556#557#558#559#560#561#562#563#564#565#566#567#568#569#570#571#572#573#574#575#576#577#578#579#580#581#582#583#584#585#586#587#588#589#590#591#592#593#594#595#596#597#598#599#600#601#602#603#604#605#606#607#608#609#610#611#612#613#614#615#616#617#618#619#620#621#622#623#624#625#626#627#628#629#630#631#632#633#634#635#636#637#638#639#640#641#642#643#644#645#646#647#648#649#650#651#652#653#654#655#656#657#658#659#660#661#662#663#664#665#666#667#668#669#670#671#672#673#674#675#676#677#678#679#680#681#682#683#684#685#686#687#688#689#690#691#692#693#694#695#696#697#698#699#700#701#702#703#704#705#706#707#708#709#710#711#712#713#714#715#716#717#718#719#720#721#722#723#724#725#726#727#728#729#730#731#732#733#734#735#736#737#738#739#740#741#742#743#744#745#746#747#748#749#750#751#752#753#754#755#756#757#758#759#760#761#762#763#764#765#766#767#768#769#770#771#772#773#774#775#776#777#778#779#780#781#782#783#784#785#786#787#788#789#790#791#792#793#794#795#796#797#798#799#800#801#802#803#804#805#806#807#808#809#810#811#812#813#814#815#816#817#818#819#820#821#822#823#824#825#826#827#828#829#830#831#832#833#834#835#836#837#838#839#840#841#842#843#844#845#846#847#848#849#850#851#852#853#854#855#856#857#858#859#860#861#862#863#864#865#866#867#868#869#870#871#872#873#874#875#876#877#878#879#880#881#882#883#884#885#886#887#888#889#890#891#892#893#894#895#896#897#898#899#900#901#902#903#904#905#906#907#908#909#910#911#912#913#914#915#916#917#918#919#920#921#922#923#924#925#926#927#928#929#930#931#932#933#934#935#936#937#938#939#940#941#942#943#944#945#946#947#948#949#950#951#952#953#954#955#956#957#958#959#960#961#962#963#964#965#966#967#968#969#970#971#972#973#974#975#976#977#978#979#980#981#982#983#984#985#986#987#988#989#990#991#992#993#994#995#996#997#998#999#1000
10 \count@\catcode#1\relax
11 \toks@\expandafter{\zen@AtEnd \catcode#1 }
12 \xdef\zen@AtEnd{\the\toks@ \the\count@\relax}}
13 \catcode#1 #2\relax
14 }
15 \TMP@EnsureCode{40}{12} % (
16 \TMP@EnsureCode{41}{12} % )
17 \TMP@EnsureCode{'\I}{14} % İ (default comment: itemize option)
18 \TMP@EnsureCode{'\D}{14} % Đ (default comment: description option)
19 \TMP@EnsureCode{'\E}{9} % Ę (default ignore: external option)

```

4.2 The package options

```

20 \SetupKeyvalOptions{family=zen,prefix=zen@}
21 \DeclareBoolOption{CaSeS}
22 \DeclareBoolOption{itemize}
23 \DeclareBoolOption{description}
24 \DeclareBoolOption{enumerate}
25 \DeclareVoidOption{greek}{\AtEndOfPackage\zen@option@greek}
26 \def\zen@option@greek{%
27 \RequirePackage{engrec,alphalph}
28 \let\zen@grecORI \@grec \let\zen@GrecORI \@Grec
29 \let\@grec \relax \let\@Grec \relax
30 \newalphalph\@grec[wrap]\zen@grecORI{24}
31 \newalphalph\@Grec[wrap]\zen@GrecORI{24}
32 \AddEnumerateCounter\engrec\@grec{\@grec{23}}% widest is \psi
33 \AddEnumerateCounter\EnGrec\@Grec{\@Grec{12}}% widest is M
34 \let\zen@option@greek \relax
35 }% \zen@option@greek
36 \DeclareVoidOption{greekctr}{\AtEndOfPackage\zen@option@greekctr}
37 \def\zen@option@greekctr{%
38 \RequirePackage{greekctr,alphalph}
39 \let\zen@greekORI \@greek \let\zen@GrecORI \@Grec
40 \let\@greek \relax \let\@Grec \relax
41 \newalphalph\@greek[wrap]\zen@greekORI{24}%
42 \newalphalph\@Grec[wrap]\zen@GrecORI{24}%
43 \AddEnumerateCounter\greek\@greek{\@greek{23}}% widest is \psi
44 \AddEnumerateCounter\Grec\@Grec{\@Grec{12}}% widest is M
45 \let\zen@option@greekctr \relax
46 }% \zen@option@greekctr
47 \DeclareVoidOption{external}{\catcode'\E 14\relax}% Ę = comment
48 \DeclareDefaultOption{\@unknownoptionerror}
49 \AtBeginDocument{%
50 \@ifpackageloaded{greekctr} \zen@option@greekctr \relax
51 \@ifpackageloaded{engrec} \zen@option@greek \relax
52 }
53 \ProcessLocalKeyvalOptions{zen}

```

```

54 \ifzen@enumerate\else
55 \ifzen@itemize \zen@enumeratetrue\else
56 \ifzen@description \zen@enumeratetrue\else
57 \zen@enumeratetrue \zen@itemizetrue \zen@descriptiontrue
58 \fi\fi
59 \ifzen@enumerate\else
60 \zen@enumeratetrue \PackageWarning{enumitem-zref}
61 {Option 'enumerate = false' has no effect}%
62 \fi
63 \fi
64 \ifzen@itemize \catcode'\I 9 \fi % ignore I
65 \ifzen@description\catcode'\D 9 \fi % ignore D
66 \AtBeginDocument{\zen@item@comma \setcounter{zenItem}\z@}

```

Babel French: `frenchb` redefines `\itemize` at begin document.

Therefore, when option `[itemize]` of `enumitem-zref` is used, one have to ensure that babel-french setup `CompactItemize=false` is set:

```
67 \I \AtEndPreamble{\ifdefined\frenchbsetup\frenchbsetup{CompactItemize=false}\fi}% [itemize] option

```

4.3 Some constants

`zenItem` is the global L^AT_EX counter for zref labels and anchors added by `enumitem-zref`

`\c@zenidx` is the local counter for the index of the items. Every item is counted.

`\zen@toks` A token to be used by `\refitem` at the time the (optional) properties are analyzed: we can't use `\toks@` at this time, because `\toks@` could have been set to a value anywhere else.

```

68 \newcounter{zenItem}
69 \globcount\c@zenidx
70 \newtoks\zen@toks

```

`\zen@@autolabel` To check if the item user-defined or automatic.

```
71 \def\zen@@autolabel{\@itemlabel}

```

`\zen@@norefence` How to display undefined references.

```
72 \def\zen@@norefence{\nfss@text{\reset@font\bfseries ??}}

```

`\zen@Hy@anchor` is used to put the anchors for the items. It's a wrapper for `hyperref` `\Hy@raisedlink`.

```

73 \protected\def\zen@Hy@anchor#1{%
74 \Hy@raisedlink{\hyper@anchorstart{#1}\hyper@anchorend}%
75 }% \zen@Hy@anchor (wrapper for \Hy@raisedlink)

```

`\zen@lowercase` is used to make lowercase references in the `refitem` scheme, when the package option `CaSeS` is not `\zen@CaSeS` active.

```

76 \let\zen@CaSeS \@firstofone
77 \ifzen@CaSeS \let\zen@lowercase \zen@CaSeS
78 \else \let\zen@lowercase \lowercase
79 \fi

```

`\zen@ifrefundefined` Switch to the first part if the L^AT_EX label exists, to the second part otherwise.

```
80 \def\zen@ifrefundefined#1{\ltx@ifundefined{r@#1}}

```

`\zen@box` A utility to expand code inside a temporary `\vbox`

```

81 \def\zen@box{\hfuzz\maxdimen \vfuzz\hfuzz \hbadness\@M \vbadness\hbadness
82 \tracinglostchars\z@ \everypar{}\setbox\z@=\vbox}

```

4.4 Properties for the zref list: zen@list

`\zen@unique` is the unique counter to create unique labels for zref references (internal use only):

```

83 \zref@newlist{zen@list}
84 \def\zen@{zen>\the\c@zenItem}
85 \zref@newprop{zen@}[\zen@
86 \def\zen@temp#1#2#3{%
87   \zref@newprop{zen@#2}[\zen@@norefrence]{#3}%
88   \ifx U#1\expandafter\let\csname zen@extract@#2\endcsname\unexpanded
89   \else \expandafter\let\csname zen@extract@#2\endcsname\detokenize
90   \fi
91 }%
92 \zref@newprop*{zen@page}[\zen@@norefrence]\thepage
93 \expandafter\let\csname zen@extract@page\endcsname\unexpanded
94 \zen@temp U{anchor}\zen@@anchor
95 \zen@temp U{default}\zen@@ref
96 \zen@temp U{type}{\ifdefined\zen@@type\ifnum\zen@@type<z@\else \enit@type \fi\fi}
97 \zen@temp D{listname}\zen@@listname
98 \zen@temp U{depth}{\the\@listdepth}
99 \zen@temp U{index}{\the\c@zenidx}
100 \zen@temp U{listctr}{\ifdefined\@listctr \the\value\@listctr\fi}
101 \zen@temp U{parents}\zen@@parents
102 \zen@temp U{label}\zen@@itemlabel
103 \zen@temp U{starlabel}\zen@@starlabel
104 \zen@temp U{ref}\zen@@ref
105 \zen@temp U{item}\@itemlabel
106 \zen@temp U{currentlabel}\@currentlabel
107 \zen@temp U{currentlabelname}\@currentlabelname
108 \zen@temp D{refitem}\zen@@refitem
109 \zen@temp D{commaref}\zen@@commaref
110 \zen@temp U{counter}{\zref@getcurrent{counter}}
111 \zref@addprop{zen@list}{%
112   zen@default,zen@page,zen@type,zen@depth,zen@listctr,%
113   zen@listname,zen@index,zen@label,zen@ref,zen@starlabel,%
114   zen@parents,zen@refitem,zen@commaref,zen@item,zen@currentlabel,zen@counter}
115 \newif\ifzen@Hy
116 \AtBeginDocument{%
117   \ifdefined\hyper@anchor \zen@Hytrue
118   \zref@addprop{zen@list}{zen@anchor,zen@currentlabelname}%
119   \else \zen@Hyfalse \let\zen@refHy \zen@refText
120   \fi
121 }

```

4.5 Normalisation of references and item labels

`\zen@Normalize`

```

122 \def\zen@Normalize#1#2#3{\begingroup \let\zen@=#1%
123 \def\GetTitleStringResult{#3}\toks@{}%
124 \ifx \GetTitleStringResult\@empty \else
125   \zen@NormalizeCommands \GetTitleStringExpand{#3}%
126   \expandafter\zen@trailingspaces \GetTitleStringResult{\@nnil}\@nnil
127 \fi
128 \edef\x{\endgroup
129   \def\noexpand#2{\expandafter\zen@zap@doublespaces
130     \detokenize\expandafter{\GetTitleStringResult} \@empty}%
131   \def\noexpand\zen@temp {\expandafter\zen@zap@doublespaces
132     \detokenize\expandafter{\the\toks@} \@empty}%
133 } \x
134 }% \zen@Normalize
135 \def\zen@trailingspaces#1#{\toks@{#1}\expandafter\zen@trailingsp@ces}
136 {\catcode47 = 8 % /
137 \gdef\zen@trailingsp@ces#1{%
138   \ifx \@nnil#1\@empty \expandafter\remove@to@nnil

```

```

139 \else \zen@{\toks@\expandafter{\the\expandafter\toks@
140         \expandafter{\romannumeral0\zen@postspace #1/ /}}}%
141     \expandafter\zen@trailingsp@ces      % (loop)
142 \fi
143 }% \zen@trailingsp@ces
144 \gdef\zen@postspace#1 /{\zen@postsp@ce#1/}
145 \gdef\zen@postsp@ce#1/#2{ #1}
146 }% \catcode group
147 \def\zen@NormalizeCommands{%
148     \let\ensuremath \@firstofone      % \ensuremath (engrec package)
149     \let\relax \@empty
150     \csname @safe@activestrue\endcsname
151 }% \zen@NormalizeCommands
152 \def\zen@zap@doublespaces#1 #2{%
153     #1%
154     \ifx#2\@empty\else\space\expandafter\zen@zap@doublespaces\fi
155     #2%
156 }% \zen@zap@doublespaces

```

4.6 Keys `name` and `name*` are added to `enumitem` environments

```

157 \define@key{enumitem}{name}[]{\zen@namelist{#1}}
158 \define@key{enumitem}{name*}[]{\zen@starnamelist{#1}}

```

`\zen@namelist` is the callback for the key `name`.

```

159 Ë \AtBeginDocument{\gappto\enit@list{,name}}% ignore if [external]
160 \def\zen@namelist #1{%
161     \ifdefined\zen@@grouplevel\else \let\zen@@grouplevel \z@ \fi
162     \ifnum\zen@@grouplevel<\currentgrouplevel% first (automatic) assignment to name
163         \zen@type                      % = (automatic from \enit@list)
164         \ifcase\zen@@type                % (case 0) = enum
165             \zen@autoname
166         \or                               % (case 1) = itemize
167     Ì         \zen@autoname                % [itemize] option only
168         \or                               % (case 2) = description
169     Ð         \zen@autoname                % [description] option only
170     \fi
171 \else \zen@Normalize\zen@CaSeS\zen@@listname {#1}%
172     \let\zen@@listname \zen@temp
173 \fi
174 }% \zen@namelist (code for key name)

```

`\zen@starnamelist` is the callback for the key `name*`.

```

175 \def\zen@starnamelist #1{% (1st automatic assignment to name done before)
176     \zen@Normalize\zen@CaSeS\zen@@listname{#1}%
177     \let\zen@@listname \zen@temp
178     \ifx \zen@@priorlistname\zen@@listname
179         \let\zen@@listname \@empty \fi
180     \zen@namelist\zen@@listname
181 }% \zen@starnamelist (code for key name*)

```

`\zen@autoname` is the callback for the key `name` automatically set without argument by `\setkeys {enit@list}`. This corresponds to some setup:

- save the prior list name *etc..*
- define the group level,
- define the list depth,
- reset the index,
- save the definition of `\@trivlist` and overload it temporarily until its next expansion at the beginning of the list.

```

182 \def\zen@autoname{% (1st automatic assignment to key name (by \enit@list))
183     \unless \ifx\zen@@listname\@empty

```

```

184   \let\zen@@priorlistname \zen@@listname \fi
185   \unless\ifdefined\zen@@priorlistname \let\zen@@priorlistname \@empty\fi
186   \let\zen@@listname \@empty
187   \ifdefined\zen@@commaref \let\zen@@priorcommaref \zen@@commaref
188   \else
189       \let\zen@@priorcommaref \@empty
190       \let\zen@@commaref \@empty
191   \fi
192   \ifdefined\zen@@refitem \let\zen@@priorrefitem \zen@@refitem
193   \else
194       \let\zen@@priorrefitem \@empty
195       \let\zen@@refitem \@empty
196   \fi
197   \edef\zen@@grouplevel {\the\currentgrouplevel}%
198   \edef\zen@@depth {\number\@listdepth}%
199   \ifnum \zen@@depth>\@one \edef\zen@@parents{\the\c@zenItem}\zen@@parents}%
200   \else
201       \let\zen@@parents \@empty \fi
202   \let\zen@refitem@fun \@undefined
203   \ifcase \zen@@type \zen@prepare@resume \fi % (enumerate only: key resume)
204   \c@zenidx \z@
205   \let\zen@trivlistORI \@trivlist
206   \let\@trivlist \zen@trivlist
207 }% \zen@name@list
208 \def\zen@prepare@resume{\expandafter\def\expandafter\enit@endenumerate
209 \expandafter{\enit@endenumerate \zen@endenumerate}%
210 }% \zen@prepare@resume
211 \def\zen@endenumerate{% saves the value of \c@zenidx at the exit of enum envir
212 \zen@toks\expandafter{\enit@afterlist}%
213 \xdef\enit@afterlist{\the\zen@toks
214 \def\expandafter\noexpand
215 \csname enit@resume@\@currenvir\endcsname{%
216 \csname c@\@listctr\endcsname
217 \the\csname c@\@listctr\endcsname
218 \c@zenidx \the\c@zenidx\relax}}}%
219 }% \zen@prepare@resume

```

`\zen@trivlist` `\@trivlist` is temporarily overloaded, and the original definition is immediately restored at the time it will be expanded, just after having set all the specified keys for the list environment.

The overload sets `\zen@setuprefs` to be expanded at the end of `\@trivlist`.

```

217 \def\zen@trivlist{%
218   \let\@trivlist \zen@trivlistORI % restore original immediately
219   \@trivlist \zen@setuprefs
220 }% \zen@trivlist

```

`\zen@type` expands to:

- 0 if enumerate
- 1 if itemize (and [itemize] option is active)
- 2 if description (and [description] option is active)
- 1 in any other case

```

221 \def\zen@type{%
222   \ifdefined\enit@type
223     \expandafter\strip@prefix \if e\enit@type>\let\zen@@type\z@ % enum begins with e
224     \else>\expandafter\strip@prefix \if i\enit@type>\let\zen@@type\@one % item begins with i
225     \else>\expandafter\strip@prefix \if d\enit@type>\let\zen@@type\tw@ % desc begins with d
226     \else>\let\zen@@type\m@ne
227     \fi\fi\fi
228   \else \let\zen@@type\m@ne
229   \fi
230 }% \zen@type

```

4.7 Overloading \makelabel

\zen@setuprefs expands just at the end of \@trivlist. \makelabel is overloaded in order to make the references and write the labels into the .aux file.

\zen@makelabel

```

231 \def\zen@setuprefs{%
232   \edef\zen@temp{\zen@priorlistname\zen@listname}%
233   \ifx\zen@temp\@empty
234     \ifx\makelabel\zen@makelabel \let\makelabel\zen@makelabelORI \fi
235   \else
236     \ifx\makelabel\zen@makelabel \else
237       \let\zen@makelabelORI\makelabel
238       \let\makelabel\zen@makelabel \fi
239   \fi
240   \let\zeninfo\zen@info
241   \ifcase\zen@ctype % enumerate only (key start)
242     \ifnum\c@zenidx=\z@ \c@zenidx\value\@enumctr\relax \fi
243   \fi
244   \zen@refitem@implicit
245 }% \zen@setuprefs
246 \def\zen@makelabel#1{\zen@makeref{#1}\zen@makelabelORI{#1}\zen@labelize}

```

\zen@refitem@implicit defines \zen@refitem@fun which cancels text around the formatted counter in enumerate lists (*ie.* numbered lists). This will be used at the time of \makelabel to create the refitem and the property \label* (which corresponds to zref property zen@starlabel).

```

247 \def\zen@refitem@implicit{% determine value of key refitem according to the label
248   \def\zen@gtemp{{\the\c@zenidx}}%
249   \ifnum\zen@ctype=\z@ % enumerate
250     {\let\enit@refstar\zen@refitemfromlabel
251     \enit@labellist \zen@box{\@itemlabel}}%
252   \fi
253   \let\zen@refitem@fun\zen@gtemp
254 }% \zen@refitem@implicit
255 \def\zen@refitemfromlabel#1{\def#1##1{0%
256   \def\zen@temp{##1}\ifx\zen@temp\@enumctr
257     \gdef\zen@gtemp{##1{##1}}\fi}%
258 }% \zen@refitemfromlabel

```

4.8 Creating references

\zen@makeref is expanded just before the original version of \makelabel

The aim is to set the zref properties defined by enumitem-zref for the current item, and to define the anchor names.

```

259 \def\zen@makeref#1{% #1 = argument of \makelabel
260   \def\zen@@itemlabel{#1}\advance\c@zenidx \@ne
261   \ifx\zen@@itemlabel\zen@@autolabel
262     \if@nmbrrlist
263       \global\advance\c@zenItem \@ne
264       \let\zen@@ref\@currentlabel
265     \else
266       {\refstepcounter{zenItem}}%
267       \let\zen@@ref\zen@refitem@fun
268     \fi
269     \let\zen@@starlabel\zen@refitem@fun
270     \zen@Normalize\zen@lowercase\zen@currentrefitem\zen@refitem@fun
271   \else % user defined label
272     {\refstepcounter{zenItem}}%
273     \zen@Normalize\zen@lowercase\zen@@ref{{\zen@@itemlabel}}%
274     \let\zen@@starlabel\zen@@ref
275   \fi
276   \let\zen@currentrefitem\zen@temp % \zen@temp defined by \zen@Normalize

```

```

277 \let\zen@@anchor \@currentHref
278 \zen@makeref@item@comma
279 }% \zen@makeref

```

`\zen@labelize` is expanded just after the original version of `\makelabel`

zlabels are written into the .aux file.

```

280 \def\zen@labelize{%
281 \ifzen@Hy \zen@anchors
282 \zref@labelbyprops{zen->\@currentHref}{zen@}\fi
283 \zref@labelbylist\zen@{zen@list}%
284 \zen@keeplabel
285 }% \zen@makeref

```

`\zen@makeref@item@comma` Creates the references for the *commaref* and the *refitem* schemes.

```

286 \def\zen@makeref@item@comma{%
287 \begingroup % to avoid silent assignment to \relax
288 \edef\x{\endgroup
289 \def\noexpand\zen@@commaref{%
290 \expandafter\zen@commaref@\zen@@priorcommaref \@nnil}%
291 \def\noexpand\zen@@refitem{%
292 \expandafter\zen@refitem@\zen@@priorrefitem \@nnil}%
293 }\x
294 }% \zen@makeref@item@comma
295 \def\zen@refitem@#1{%
296 \ifx \@nnil#1%
297 \ifx \zen@@listname \@empty \else
298 {\zen@@listname\zen@currentrefitem}\fi
299 \else {#1\zen@currentrefitem}\expandafter\zen@refitem@ % loop
300 \fi
301 }% \zen@refitem@
302 \def\zen@commaref@#1{%
303 \ifx \@nnil#1%
304 \ifx \zen@@listname \@empty \else
305 {\zen@@listname,\the\c@zenidx}\fi
306 \else {#1,\the\c@zenidx}\expandafter\zen@commaref@ % loop
307 \fi
308 }% \zen@commaref@

```

`\zen@anchors` Put the PDFanchors for the current item.

```

309 \let\zen@list@anchors \@empty
310 \def\zen@anchors{\begingroup
311 \def\zen@@anchors{\endgroup}%
312 \expandafter\expandafter\expandafter\zen@anchors@
313 \expandafter\zen@@commaref \zen@@refitem \@nnil
314 \zen@@anchors
315 }% \zen@anchors
316 \def\zen@anchors@#1{%
317 \ifx \@nnil#1\else
318 \expandafter\in@\csname refitem.#1\expandafter\endcsname\expandafter{\zen@list@anchors}%
319 \ifin@ \zen@anchor@ignore{#1}%
320 \else
321 \edef\zen@@anchors{\zen@@anchors\zen@Hy@anchor{refitem.#1}}%
322 \xdef\zen@list@anchors{%
323 \unexpanded\expandafter{\zen@list@anchors}%
324 \expandafter\noexpand\csname refitem.#1\endcsname}%
325 \fi
326 \expandafter\zen@anchors@ % loop
327 \fi
328 }% \zen@anchors@ (loop)
329 \def\zen@anchor@ignore#1{\PackageInfo{enumitem-zref}{Ignored anchor '#1'}}

```

`\zen@keeplabel` As far as references are built and written inside `\makelabel` (which was overloaded) they are built inside a group (inside a `\sbox`, and thus inside a `\color@setgroup... \color@endgroup` group if packages

color or xcolor are used, and may be inside other grouping levels depending on the user need (she may have redefined `\makelabel` for a special purpose).

Therefore, we take advantage of the e-TeX command `\currentgrouplevel` in order to decide how many levels of grouping we have to “pass through” for keeping the current values of `\zen@zenidx` (the item *index*), `\zen@@refitem`, and `\zen@@commaref`.

The “target” group level has been saved into `\zen@@grouplevel` at the beginning of the environment, while setting the automatic `name` key for the first time.

```

330 \def\zen@keep#1{\def\noexpand#1{\unexpanded\expandafter{#1}}}%
331 \def\zen@keeplabel{%
332   \xdef\zen@gtemp{%
333     \noexpand\ifnum \zen@@grouplevel<\currentgrouplevel
334       \aftergroup \noexpand\zen@gtemp
335     \noexpand\else
336       \c@zenidx \the\c@zenidx\relax
337       \zen@keep \zen@@refitem
338       \zen@keep \zen@@commaref
339       \ifzen@Hy \zen@keep\zen@@anchor\fi
340     \noexpand\fi
341   }\aftergroup\zen@gtemp
342 }% \zen@keeplabel

```

4.9 Extracting zref properties

`\zen@extract` A wrapper to `\zref@extractdefault` which can also `\detokenize` the result.

```

343 \newcommand*\zen@extract[3]{% #1=ref #2=prop #3=def
344   \csname zen@extract@#2\endcsname\expandafter\expandafter%
345   \expandafter{\zref@extractdefault{zen>\number#1}{zen@#2}{#3}}%
346 }% \zen@extract

```

4.10 Referencing the items: `\refitem`

`\refitem` The general macro to print references and insert hyperlinks, with a star variant (no hyperlink).

```

347 \newrobustcmd*\refitem{\begingroup \ifstar
348   {\zen@refitemOpt \zen@refText }%      no hyperlink
349   {\zen@refitemOpt \zen@refHy }%
350 }% \refitem
351 \def\zen@refHy{%
352 %%   \edef\zen@@anchor{\zen@extract\c@zenItem{anchor}{}\unexpanded}%
353 %%   \ifx \zen@@anchor\@empty \zen@@text
354 %%   \else
355     \hyperref{}{}{\zref@extract{zen>\the\c@zenItem}{zen@anchor}}\zen@@text
356 %%   \fi
357 }% \zen@ref
358 \def\zen@refText{\zen@@text}

```

`\zen@refitemOpt` If the first argument is an option, then `\refitem` makes a hyperlink to an external document.

```

359 \def\zen@refitemOpt#1{\let\zen@ref@#1\ifnextchar\zen@refitemExternal \zen@refitemRef }
360 \def\zen@refitemRef #1{\@tempwatrue \let\zen@lastwarn \m@ne
361   \zen@Normalize\zen@lowercase\zen@@label {#1}%
362   \zen@ifrefundefined \zen@@label
363   {\expandafter\expandafter\expandafter \zen@refused
364     \csname\ifcsname zen@refitem(\zen@temp)\endcsname
365       zen@refitem(\zen@temp)\else
366       zen@norefitem\fi\endcsname>}%
367   \zen@refitemFromAnchor
368 }% \zen@refitemRef
369 \def\zen@norefitem {z@>0}

```

`\zen@refused` Checks if the label exists in the *commaref* scheme or in the *refitem* scheme.

```

370 \def\zen@refused#1>#2>{%

```



```

371 \ifcase#1\relax
372   \@tempswafalse \refused\zen@@label
373   \let\zen@ref@ \zen@refText % do not create a hyperlink if no reference
374 \or \let\zen@@label \zen@temp \c@zenItem =#2\relax
375 \else \let\zen@@label \zen@temp \zen@MultipleLabels
376   \c@zenItem =#2\relax
377 \fi
378 \zen@refitem@prop
379 }% \zen@refused

```

`\zen@refitemFromAnchor` When `\refitem` refers to a user item, we can reached the zref properties with the chain:

LaTeX label \longrightarrow hyperref anchor \longrightarrow enumitem-zref label \longrightarrow property

```

380 \def\zen@refitemFromAnchor{%
381   \edef\zen@@anchor{\getrefbykeydefault\zen@@label{anchor}{}}%
382   \ifx \zen@@anchor\@empty
383     \@tempswafalse \G@refundefinedtrue \zen@warn\z@
384     \let\zen@ref@ \zen@refText
385   \else
386     \edef\zen@temp{\zref@extractdefault{zen->\zen@@anchor}{zen@}{}}%
387     \ifx \zen@temp\@empty
388       \@tempswafalse \G@refundefinedtrue \zen@warn\@ne
389       \ifzen@Hy \ifx\zen@ref@ \zen@refHy \def\zen@ref@{\ref{\zen@@label}}%
390         \else \def\zen@ref@ {\ref*{\zen@@label}}%
391       \fi
392     \else \def\zen@ref@{\ref{\zen@@label}}%
393     \fi
394   \else \c@zenItem \expandafter\strip@prefix\zen@temp\relax
395   \fi
396 \fi
397 \zen@refitem@prop
398 }% \zen@refitemFromAnchor

399 \def\zen@warn#1{\let\zen@lastwarn#1%
400 \message{Package enumitem-zref Warning:
401 \ifcase#1 %
402   \string\refitem{\zen@@label}:
403   User references to \string\label\space^^J\zen@spaces
404   require hyperref and the appropriate options for enumitem-zref^^J\zen@spaces
405   \ifzen@Hy Check the options given to package enumitem-zref.
406   \else Package hyperref is not loaded.\fi
407 \or User reference does not point to an item!^^J\zen@spaces
408   \string\refitem{\zen@@label}
409 \fi\on@line^^J^^J}}
410 \edef\zen@spaces{\@spaces\@spaces}

```

`\zen@refitem@prop` tests if `\refitem` has a following optional argument: the properties.

As long as the optional argument is in last position, we have to take a special care about the the spaces (`\@ifnextchar` removes spaces).

```

411 \def\zen@refitem@prop{\futurelet\zen@temp \zen@refitemProp}
412 \def\zen@gobblespace#1 {#1}%
413 \let\zen@space \relax
414 \def\zen@refitemProp{%
415   \ifcase 0\ifx \zen@temp\@else
416     \ifx \zen@temp\@sptoken 1\else
417     2\fi\fi\relax
418     \expandafter \zen@refitemPrint
419   \or \let\zen@space\space
420     \expandafter\zen@gobblespace\expandafter\zen@refitem@prop
421   \else \expandafter\zen@refitemPrint \expandafter[\expandafter]%
422   \fi
423 }% \zen@refitemProp

```

`\zen@refitemPrint` is the final macro to print the reference made by `\refitem`. The properties, if specified, are first analyzed by `\zen@refitemText`.

```

424 \long\def\zen@refitemPrint[#1]{\def\zen@@text{#1}%
425 \if@tempswa % got a reference
426 \ifx \zen@@text\@empty
427 \edef\zen@@text{\zen@extract\c@zenItem{default}\zen@@noreference}\zen@space}%
428 \else\zen@refitemText
429 \fi
430 \else
431 \ifx \zen@@text\@empty \def\zen@@text{\zen@@noreference\zen@space}%\refused\zen@@label
432 \else \zen@refitem@noText
433 \fi
434 \fi
435 \zen@ref@
436 \endgroup
437 }% \zen@refitemPrint

```

`\zen@refitemText` creates the text to be typeset for the reference, depending on the properties given by the user as the last optional argument to `\refitem`.

```

438 \def\zen@refitemText{\zen@propcount
439 \ifnum \c@zenidx>\z@ \advance\c@zenidx \m@ne
440 \ifnum \c@zenidx>\z@
441 \edef\zen@@parents{\zen@extract\c@zenItem{parents}{}}%
442 \edef\zen@@depth{\zen@extract\c@zenItem{depth}0}%
443 \else \let\zen@@depth \@ne % shortcut (only one property)
444 \fi
445 \def\ref{\@ifstar{\zen@ancestorprop{starlabel}}{\zen@ancestorprop{ref}}}%
446 \def\label{\@ifstar{\zen@ancestorprop{starlabel}}{\zen@ancestorprop{label}}}%
447 \def\default{\zen@ancestorprop{default}}\def\counter{\zen@ancestorprop{counter}}%
448 \def\type{\zen@ancestorprop{type}}\def\listname{\zen@ancestorprop{listname}}%
449 \def\index{\zen@ancestorprop{index}}\def\listctr{\zen@ancestorprop{listctr}}%
450 \def\depth{\zen@ancestorprop{depth}}\def\refitem{\zen@ancestorprop{refitem}}%
451 \def\anchor{\zen@ancestorprop{anchor}}\def\commaref{\zen@ancestorprop{commaref}}%
452 \def\page{\zen@ancestorprop{page}}\def\currentlabel{\zen@ancestorprop{currentlabel}}%
453 \def\currentlabelname{\zen@ancestorprop{currentlabelname}}%
454 {\zen@box{\zen@@text
455 \expandafter\expandafter\expandafter\zen@toks
456 \expandafter{\the\zen@toks \zen@nil}%
457 \def\ref{\the\zen@toks}\zen@clprops
458 \def\ref{\@ifstar{\the\zen@toks}{\the\zen@toks}}\let\label\ref
459 \long\def\zen@##1##2\zen@nil{\zen@toks{##2\zen@nil}##1}\let\zen@nil\relax
460 \fi
461 }% \zen@refitemText
462 \def\zen@ancestorprop#1{%
463 \ifnum \zen@@depth>\c@zenidx
464 \advance\c@zenidx \m@ne
465 \ifnum \c@zenidx<\z@ \let\zen@ \c@zenItem
466 \else \edef\zen@\{\zen@ancestor}%
467 \fi
468 \edef\zen@temp{\zen@extract\zen@{#1}\zen@@noreference}}%
469 \else \def\zen@temp{\}\advance\c@zenidx \m@ne
470 \fi
471 \zen@toks\expandafter{\the\expandafter\zen@toks \expandafter\zen@ \zen@temp}%
472 }% \zen@ancestorproperty
473 \def\zen@ancestor{\expandafter\ltx@car\romannumeral0\expandafter\ltx@GobbleNum
474 \expandafter{\the\expandafter\c@zenidx\expandafter}\zen@@parents{}}{\}\@nil % no need of {}...
475 }% \zen@getancestor
476 \def\zen@propcount{\c@zenidx\z@
477 \def\ref{\advance\c@zenidx \@ne}\zen@clprops
478 {\zen@box{\zen@@text \expandafter\expandafter\expandafter
479 \c@zenidx \the\c@zenidx\relax
480 }% \zen@propcount
481 \def\zen@clprops{\let\label\ref \let\page\ref
482 \let\none\ref \let\default\ref \let\counter\ref \let\listname\ref

```

```

483 \let\anchor\ref \let\depth\ref \let\type\ref \let\index\ref \let\listctr\ref
484 \let\refitem\ref \let\commaref\ref \let\currentlabel\ref \let\currentlabelname\ref
485 }% \zen@clprops
486 \def\zen@refitem@noText{\let\ref \zen@@noreference \zen@clprops}

```

`\zen@refitemExternal` for references to an anchor in an external document with `\hyperref`:

```

487 \def\zen@refitemExternal[#1]#2{%
488 \toks@{#1}\zen@Normalize\zen@lowercase\zen@@label {#2}%
489 \edef\zen@ref@{\endgroup
490 \noexpand\hyperref{\the\toks@}{refitem}{\zen@temp}%
491 }\zen@ref@
492 }% \zen@Hy@external

```

4.11 Undefined references and duplicate labels management

`\zen@MultipleLabels`

```

493 \def\zen@MultipleLabels{%
494 \PackageWarning{enumitem-zref}
495 {'\zen@@label' on page \thepage \space multiple defined\MessageBreak
496 Only the first one will be referenced}% et mettre ici la liste des références correspondant aux alias
497 }% \zen@MultipleLabels

```

4.12 `\zeninfo` (If you are lost...)

`\zen@info` `\zeninfo` executes `\zen@info` when inside environments:

```

498 \newrobustcmd*\zen@info[1][\begingroup \sloppy
499 \def\sep##1{##1\hskip\z@ plus.001fil\relax\linebreak[2]}%
500 \ifcase\zen@whichinfo{#1} \zen@@depth % case 0 (depth)
501 \or \the\c@zenidx % case 1 (index)
502 \or % case 2 (type)
503 \ifcase\zen@@type enumerate\or itemize\or description\or unknown\fi
504 \or anchor=\ifzen@Hy\zen@@anchor\else no hyperref\fi % case 3 (anchor)
505 \or refitem=\sep{}\detokenize\expandafter{\zen@refitem}% case 4 (refitem)
506 \or commaref=\sep{}\detokenize\expandafter{\zen@@commaref}% case 5 (commaref)
507 \else % (case -1) give all infos
508 {\ifdefined\scriptsize\scriptsize\fi
509 zeninfo:\sep{}depth=\sep{}\zen@@depth
510 \sep/ idx=\sep{}\the\c@zenidx
511 \ifcase\zen@@type \sep/ ctr=\sep{}\the\value\@listctr \fi % enumerate only
512 \sep/ commaref=\sep{}\detokenize\expandafter{\zen@@commaref}%
513 \sep/ refitem=\sep{}\detokenize\expandafter{\zen@refitem}%
514 \ifzen@Hy\sep/ anchor=\sep{}\zen@@anchor \fi
515 }%
516 \fi\endgroup
517 }% \zen@info

```

`\zeninfo` is executed when out of scope:

```

518 \newrobustcmd\zeninfo[1][\begingroup
519 \ifdefined\zen@@type
520 \ifcase \zen@@type \zeninfo@err
521 I \or \zeninfo@err
522 D \or \zeninfo@err
523 \else\zeninfo@err[{in environment \@currentenv}\MessageBreak
524 You should be in a list environment}]%
525 \fi\endgroup
526 }% \zeninfo (out of scope)
527 \newcommand*\zeninfo@err[1][you should give a name to your \@currentenv\space list]}{%
528 \PackageError{enumitem-zref}
529 {\string\zeninfo\space is not available\MessageBreak
530 #1}\@eha
531 }% \zeninfo@err

```

\zen@whichinfo

```

532 \def\zen@whichinfo#1{%
533   \expandafter\strip@prefix   \if d#1>\z@   % depth (begins with d)
534   \else>\expandafter\strip@prefix\if i#1>\@ne % index (begins with i)
535   \else>\expandafter\strip@prefix\if t#1>\tw@ % type (begins with t)
536   \else>\expandafter\strip@prefix\if a#1>\thr@@% anchor (begins with a)
537   \else>\expandafter\strip@prefix\if r#1>4   % refitem(begins with r)
538   \else>\expandafter\strip@prefix\if c#1>5   % commaref(begins with c)
539   \else>\m@ne
540   \fi\fi\fi\fi\fi\fi
541 }% \zen@whichinfo

```

4.13 Create the references At Begin Document for the *refitem* and the *commaref* schemes

\zen@item@comma

```

542 \def\zen@item@comma{%
543   \begingroup \c@zenidx \@ne %%\loggingall
544   \@whilesw \ifcsname Z@R@zen>\the\c@zenidx\endcsname\fi
545     {\advance\c@zenidx \@ne}%
546   \@whilenum \c@zenidx>\@ne
547   \do{\advance \c@zenidx \m@ne
548     \zref@wrapper@unexpanded{%
549       \edef\zen@refitem{%
550         \zref@extractdefault{zen>\the\c@zenidx}{zen@refitem}{}}%
551         \zref@extractdefault{zen>\the\c@zenidx}{zen@commaref}{}}%
552       \edef\zen@refitem{\expandafter
553         \zen@item@comma@ \zen@refitem \@nnil}%
554         \expandafter\zen@def@item@comma@ \zen@refitem \@nnil}}%
555   \endgroup
556 }% \zen@item@comma
557 \def\zen@item@comma@ #1{%
558   \ifx\@nnil#1\else
559     \expandafter\noexpand\csname zen@refitem(\detokenize{#1})\endcsname
560     \expandafter\zen@item@comma@ % (loop)
561   \fi
562 }% \zen@item@comma@

```

\zen@def@item@comma@

```

563 \def\zen@def@item@comma@ #1{\ifx\@nnil#1\else
564   \zen@def@item@comma@{#1}\expandafter\zen@def@item@comma@ \fi % loop
565 }% \zen@def@item@comma@

```

\zen@def@item@comma

```

566 \newif\ifzen@duplicatealiases
567 \def\zen@def@item@comma #1{%
568   \xdef#1{%
569     \ifdefined#1%
570       \ifx\relax#1\@empty   \@ne>\the\c@zenidx% create new
571       \else \expandafter
572         \ifnum\expandafter\strip@prefix#1=\c@zenidx% redefinition to the same location
573         #1%
574       \else
575         \number\numexpr1+#1% increment multiplicity
576       \fi
577     \fi
578     \else   \@ne>\the\c@zenidx% create new
579     \fi}%
580 \ifzen@duplicatealiases\else
581 \expandafter\expandafter\expandafter\strip@prefix
582 \expandafter\strip@prefix\ifnum\@ne<#1>\global\zen@duplicatealiasestrue\else>>\fi
583 \fi

```

```
584 }% \zen@def@item@comma
585 \zen@AtEnd \let\zen@AtEnd\@undefined
586 \endpackage
```

References

- [1] *The enumitem package*; Javier Bezos
2009/05/18 v2.2 – Customized lists
CTAN:help/Catalogue/entries/enumitem.html
- [2] *The zref package*; Heiko Oberdiek
2010/05/01 v2.17 – New reference scheme for LaTeX2e
CTAN:help/Catalogue/entries/zref.html
- [3] *The gettitlestring package*; Heiko Oberdiek
2009/12/18 v1.3 – Cleanup title references
CTAN:help/Catalogue/entries/gettitlestring.html
- [4] *The engrec package*; Yvon Henel
2008/05/07 v1.1 – Greek letters from counters
CTAN:help/Catalogue/entries/engrec.html

History

[2011/02/18 v1.8]

- Documentation recompiled after `tabu`⁹ package v2.5 release.

[2010/12/30 v1.75]

- Bug fixed with empty items and nested lists.

[2010/12/27 v1.7]

- Documentation improvement.
- No modification in the package file.

[2010/12/17 v1.5]

- `enumitem-zref` now works with the keys: `start=`, `resume`, and `resume*` of `enumitem` package.
- Package option: `greekctr` added as an alternative to the `greek` package option: `greekctr` is based on the `greekctr`¹⁰ package while `greek` is based on `engrec`¹¹. Both uses code of `alphalph`¹² in order to make the counters *wrap* around the greek alphabet.
Options `greek` and/or `greekctr` are automatically loaded `\AtBeginDocument` if the `engrec` / `greekctr` packages are detected.

[2010/12/10 v1.2]

- Bug encountered in `\zen@anchors@`:
`\ifx#1\@nnil` is not the same as `\ifx \@nnil#1` !!

[2010/12/02 v1.1]

- The first version.

9. `tabu`: CTAN:help/Catalogue/entries/tabu.html

10. `greekctr`: CTAN:help/Catalogue/entries/greekctr.html

11. `engrec`: CTAN:help/Catalogue/entries/engrec.html

12. `alphalph`: CTAN:help/Catalogue/entries/alphalph.html

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	F
<code>\@Grec</code>	<code>\frenchbsetup</code> 67
<code>\@Greek</code>	<code>\futurelet</code> 411
<code>\@currentHref</code>	G
<code>\@currentlabel</code>	<code>\G@refundefinedtrue</code> 383, 388
<code>\@currentlabelname</code>	<code>\getrefbykeydefault</code> 381
<code>\@currenvir</code>	<code>\GetTitleStringDisableCommands</code> 7
<code>\@enumctr</code>	<code>\GetTitleStringExpand</code> 125
<code>\@grec</code>	<code>\GetTitleStringResult</code> 123, 124, 126, 130
<code>\@greek</code>	<code>\Greek</code> 44
<code>\@ifpackageloaded</code>	<code>\greek</code> 43
<code>\@itemlabel</code>	H
<code>\@listdepth</code>	<code>\hbadness</code> 81
<code>\@trivlist</code>	<code>\hfuzz</code> 81
<code>\@whilenum</code>	<code>\Hy@raisedlink</code> 74, 75
<code>\@whilesw</code>	<code>\hyper@anchor</code> 117
A	<code>\hyper@anchorend</code> 74
<code>\AddEnumerateCounter</code>	<code>\hyper@anchorstart</code> 74
<code>\aftergroup</code>	<code>\hyperref</code> 355, 490
<code>\anchor</code>	I
<code>\AtEndOfPackage</code>	<code>\if@nmbrlist</code> 262
<code>\AtEndPreamble</code>	<code>\ifcsname</code> 364, 544
C	<code>\ifn@</code> 319
<code>\c@zenidx</code>	<code>\ifzen@CaSeS</code> 77
68, 99, 201, 208, 215,	<code>\ifzen@description</code> 56, 65
242, 248, 260, 305, 306, 336, 439, 440, 463,	<code>\ifzen@duplicatealiases</code> 566, 580
464, 465, 469, 474, 476, 477, 479, 501, 510,	<code>\ifzen@enumerate</code> 54, 59
543, 544, 545, 546, 547, 550, 551, 570, 572, 578	<code>\ifzen@Hy</code> 115, 281, 339, 389, 405, 504, 514
<code>\c@zenItem</code>	<code>\ifzen@itemize</code> 55, 64
84, 197,	<code>\in@</code> 318
263, 352, 355, 374, 376, 394, 427, 441, 442, 465	<code>\index</code> 449, 483
<code>\commaref</code>	L
<code>\count@</code>	<code>\label</code> 403, 446, 458, 481
<code>\counter</code>	<code>\listctr</code> 449, 483
<code>\currentgrouplevel</code>	<code>\listname</code> 448, 482
<code>\currentlabel</code>	<code>\loggingall</code> 543
<code>\currentlabelname</code>	<code>\lowercase</code> 78
D	<code>\ltx@car</code> 473
<code>\DeclareBoolOption</code>	<code>\ltx@GobbleNum</code> 473
<code>\DeclareDefaultOption</code>	<code>\ltx@ifUndefined</code> 80
<code>\DeclareVoidOption</code>	M
<code>\default</code>	<code>\makelabel</code> 234, 236, 237, 238, 259
<code>\define@key</code>	<code>\message</code> 400
<code>\depth</code>	N
<code>\detokenize</code>	<code>\newalphalph</code> 30, 31, 41, 42
E	<code>\newcounter</code> 68
<code>\EnGrec</code>	<code>\newtoks</code> 70
<code>\engrec</code>	<code>\nfss@text</code> 72
<code>\enit@list</code>	<code>\none</code> 482
<code>\enit@afterlist</code>	<code>\number</code> 196, 345, 575
<code>\enit@endenumerate</code>	<code>\numexpr</code> 575
<code>\enit@format</code>	O
<code>\enit@labellist</code>	<code>\on@line</code> 409
<code>\enit@refstar</code>	
<code>\enit@type</code>	
<code>\ensuremath</code>	
<code>\everypar</code>	

P	
<code>\PackageError</code>	528
<code>\PackageInfo</code>	329
<code>\PackageWarning</code>	60, 494
<code>\page</code>	452, 481
<code>\ProcessLocalKeyvalOptions</code>	53
R	
<code>\ref</code>	389, 390, 392, 445, 457, 458, 477, 481, 482, 483, 484, 486
<code>\refitem</code>	5, 6, <u>347</u> , 402, 408, 450, 484
<code>\refstepcounter</code>	266, 272
<code>\refused</code>	372, 431
<code>\remove@to@nnil</code>	138
<code>\romannumeral</code>	140, 473
S	
<code>\setcounter</code>	66
<code>\strip@prefix</code>	223, 224, 225, 394, 533, 534, 535, 536, 537, 538, 572, 581, 582
T	
<code>\thepage</code>	92, 495
<code>\tracinglostchars</code>	82
<code>\type</code>	448, 483
V	
<code>\value</code>	100, 242, 511
<code>\vbadness</code>	81
<code>\vbox</code>	82
<code>\vfu</code>	81
Z	
<code>\zen@</code> 84, 85, 122, 139, 283, 459, 465, 466, 468, 471	
<code>\zen@@anchor</code>	94, 277, 339, 352, 353, 381, 382, 386, 504, 514
<code>\zen@@anchors</code>	311, 314, 321
<code>\zen@@autolabel</code>	<u>71</u> , 261
<code>\zen@@commaref</code> 109, 187, 189, 289, 313, 338, 506, 512	
<code>\zen@@depth</code> . . .	196, 197, 442, 443, 463, 500, 509
<code>\zen@@grouplevel</code>	161, 162, 195, 333
<code>\zen@@itemlabel</code>	102, 260, 261, 273
<code>\zen@@label</code>	361, 362, 372, 374, 375, 381, 389, 390, 392, 402, 408, 431, 488, 495
<code>\zen@@listname</code> . . .	97, 171, 172, 176, 177, 178, 179, 180, 183, 184, 186, 232, 297, 298, 304, 305
<code>\zen@@noreferrer</code> . .	<u>72</u> , 87, 92, 427, 431, 468, 486
<code>\zen@@parents</code>	101, 197, 198, 441, 474
<code>\zen@@priorcommaref</code>	187, 188, 290
<code>\zen@@priorlistname</code>	178, 184, 185, 232
<code>\zen@@priorrefitem</code>	191, 192, 292
<code>\zen@@ref</code>	95, 104, 264, 267, 273, 274
<code>\zen@@refitem</code> 108, 191, 193, 291, 313, 337, 505, 513	
<code>\zen@@starlabel</code>	103, 269, 274
<code>\zen@@text</code> 353, 355, 358, 424, 426, 427, 431, 454, 478	
<code>\zen@@type</code>	96, 164, 200, 223, 224, 225, 226, 228, 241, 249, 503, 511, 519, 520
<code>\zen@ancestor</code>	466, 473
<code>\zen@ancestorprop</code>	445, 446, 447, 448, 449, 450, 451, 452, 453, 462
<code>\zen@ancestorproperty</code>	472
<code>\zen@anchor@ignore</code>	319, 329
<code>\zen@anchors</code>	281, <u>309</u>
<code>\zen@anchors@</code>	312, 316, 326, 328
<code>\zen@autoname</code>	165, 167, 169, <u>182</u>
<code>\zen@box</code>	<u>81</u> , 251, 454, 478
<code>\zen@CaSeS</code>	<u>76</u> , 171, 176
<code>\zen@clprops</code>	457, 477, 481, 485, 486
<code>\zen@commaref@</code>	290, 302, 306, 308
<code>\zen@currentrefitem</code>	270, 276, 298, 299
<code>\zen@def@item@comma</code>	564, <u>566</u>
<code>\zen@def@item@comma@</code>	554, <u>563</u>
<code>\zen@descriptiontrue</code>	57
<code>\zen@duplicatealiasestrue</code>	582
<code>\zen@endenumerate</code>	206, 208
<code>\zen@enumeratetrue</code>	55, 56, 57, 60
<code>\zen@extract</code>	<u>343</u> , 352, 427, 441, 442, 468
<code>\zen@getancestor</code>	475
<code>\zen@gobblespace</code>	412, 420
<code>\zen@GrecORI</code>	28, 31
<code>\zen@grecORI</code>	28, 30
<code>\zen@GreekORI</code>	39, 42
<code>\zen@greekORI</code>	39, 41
<code>\zen@gtemp</code>	248, 253, 257, 332, 334, 341
<code>\zen@Hy@anchor</code>	<u>73</u> , 321
<code>\zen@Hy@external</code>	492
<code>\zen@Hyfalse</code>	119
<code>\zen@Hytrue</code>	117
<code>\zen@ifrefundefined</code>	<u>80</u> , 362
<code>\zen@info</code>	240, <u>498</u>
<code>\zen@item@comma</code>	66, <u>542</u>
<code>\zen@item@comma@</code>	553, 557, 560, 562
<code>\zen@itemizetrue</code>	57
<code>\zen@keep</code>	330, 337, 338, 339
<code>\zen@keeplabel</code>	284, <u>330</u>
<code>\zen@labelize</code>	246, <u>280</u>
<code>\zen@lastwarn</code>	360, 399
<code>\zen@list@anchors</code>	309, 318, 322, 323
<code>\zen@lowercase</code>	<u>76</u> , 270, 273, 361, 488
<code>\zen@makelabel</code>	<u>231</u>
<code>\zen@makelabelORI</code>	234, 237, 246
<code>\zen@makeref</code>	246, <u>259</u> , 285
<code>\zen@makeref@item@comma</code>	278, <u>286</u>
<code>\zen@MultipleLabels</code>	375, <u>493</u>
<code>\zen@name@list</code>	204
<code>\zen@namelist</code>	157, <u>159</u> , 180
<code>\zen@nil</code>	456, 459
<code>\zen@norefitem</code>	369
<code>\zen@Normalize</code> <u>122</u> , 171, 176, 270, 273, 276, 361, 488	
<code>\zen@NormalizeCommands</code>	125, 147, 151
<code>\zen@option@greek</code>	25, 26, 34, 35, 51
<code>\zen@option@greekctr</code>	36, 37, 45, 46, 50
<code>\zen@postsp@ce</code>	144, 145
<code>\zen@postspace</code>	140, 144
<code>\zen@prepare@resume</code>	200, 205, 207, 216
<code>\zen@propcount</code>	438, 476, 480
<code>\zen@ref</code>	357
<code>\zen@ref@</code> 359, 373, 384, 389, 390, 392, 435, 489, 491	
<code>\zen@refHy</code>	119, 349, 351, 389
<code>\zen@refitem</code>	549, 552, 553, 554
<code>\zen@refitem@</code>	292, 295, 299, 301
<code>\zen@refitem@fun</code>	199, 253, 267, 269, 270
<code>\zen@refitem@implicit</code>	244, <u>247</u>
<code>\zen@refitem@noText</code>	432, 486
<code>\zen@refitem@prop</code>	378, 397, <u>411</u>
<code>\zen@refitemExternal</code>	359, <u>487</u>
<code>\zen@refitemFromAnchor</code>	367, <u>380</u>
<code>\zen@refitemfromlabel</code>	250, 255, 258
<code>\zen@refitemOpt</code>	348, 349, <u>359</u>
<code>\zen@refitemPrint</code>	418, 421, <u>424</u>
<code>\zen@refitemProp</code>	411, 414, 423

<code>\zen@refitemRef</code>	359, 360, 368	<code>\zen@whichinfo</code>	500, 532
<code>\zen@refitemText</code>	428, 438	<code>\zen@zap@doublespaces</code>	129, 131, 152, 154, 156
<code>\zen@refText</code>	119, 358, 373, 384	<code>\zeninfo</code>	7, 240, 518
<code>\zen@reftext</code>	348	<code>\zeninfo@err</code>	520, 521, 522, 523, 527, 531
<code>\zen@refused</code>	363, 370	<code>\zenItem</code>	68
<code>\zen@setuprefs</code>	219, 231	<code>\zref@addprop</code>	111, 118
<code>\zen@space</code>	413, 419, 427, 431	<code>\zref@extract</code>	355
<code>\zen@spaces</code>	403, 404, 407, 410	<code>\zref@extractdefault</code>	345, 386, 550, 551
<code>\zen@starnamelist</code>	158, 175	<code>\zref@getcurrent</code>	110
<code>\zen@temp</code>	86, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 131, 172, 177, 232, 233, 256, 276, 364, 365, 374, 375, 386, 387, 394, 411, 415, 416, 468, 469, 471, 490	<code>\zref@labelbylist</code>	283
<code>\zen@toks</code>	68 , 209, 210, 455, 456, 457, 458, 459, 471	<code>\zref@labelbyprops</code>	282
<code>\zen@trailingsp@ces</code>	135, 137, 141, 143	<code>\zref@newlist</code>	83
<code>\zen@trailingspaces</code>	126, 135	<code>\zref@newprop</code>	85, 87, 92
<code>\zen@trivlist</code>	203, 217	<code>\zref@wrapper@unexpanded</code>	548
<code>\zen@trivlistORI</code>	202, 218		
<code>\zen@type</code>	163, 221		
<code>\zen@unique</code>	83		
<code>\zen@warn</code>	383, 388, 399		
		Ë	
		<code>\Ë</code>	19, 47
		İ	
		<code>\İ</code>	17, 64
		Ð	
		<code>\Ð</code>	18, 65