

Extended references to items for enumitem package

FC

2010/12/17 – version 1.5

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 X.

Contents

Summary of the properties for <code>\refitem</code>	2	4.3	Some constants	10
1 Forewords	3	4.4	Properties for the zref list: <code>zen@list</code> . . .	11
1.1 Numbered lists, not numbered lists		4.5	Normalisation of references and item labels	11
and hyperref	3	4.6	Keys <code>name</code> and <code>name*</code> are added to enu-	
1.2 A “natural” reference scheme for lists	3		mitem environments	12
1.2.1 The <code>commaref</code> scheme		4.7	Overloading <code>\makelabel</code>	13
1.2.2 The <code>refitem</code> scheme		4.8	Creating references	14
1.2.3 The <code>user</code> scheme: with the		4.9	Extracting zref properties	16
<code>\label</code> command		4.10	Referencing the items: <code>\refitem</code>	16
2 User Interface	5	4.11	Undefined references and duplicate labels	
2.1 Naming lists	5		management	19
2.2 Referencing the items : <code>\refitem</code> .	5	4.12	<code>\zeninfo</code> (If you are lost...)	19
2.3 The package options	6	4.13	Create the references At Begin Document	
2.4 If your are lost <code>\zeninfo</code> is your friend	7		for the <code>refitem</code> and the <code>commaref</code> schemes	20
3 Flow chart of expansion: How does	8	5 References	21	
it work ?	8	6 History	21	
4 IMPLEMENTATION	9	[2010/12/17 v1.5]	21	
4.1 Identification and requirements	9	[2010/12/10 v1.2]	21	
4.2 The package options	9	[2010/12/02 v1.1]	21	
		7 Index	21	

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]

reference with either:

The *refitem* scheme: listname{label}-{label}-{label}...

The *commaref* scheme: listname,index,index,index,...

The *user* scheme: reference to a \label command

Optional text containing properties

Default is [\ref]

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

[\ref \ref \ref]

↑

refers to the second ancestor

Here an empty string

↑

refers to the parent

1.

↑

refers to the item

ii)

1. One

i) sub item

ii) sub item

2. Two

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

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 affraid 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:

<code>\refitem{myList,2,1}</code>	this prints: 1 for it is the first item in the inner list.
<code>\refitem{myList,2,1}[\ref.\ref]</code>	prints: 2.1.
<code>\refitem{myList,2,1}[\label\ \label]</code>	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:

<code>\refitem{myList{B}{ii}{\alpha}}</code>	α
<code>\refitem{myList{B}{ii}{\alpha}}[\label\ \label\ \label]</code>	B) ii) α
<code>\refitem{myList{B}{ii}{\alpha}}[\label*---\label*---\label]</code>	B–ii– α

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}`

<code>\refitem{myList{2}{first description label}}</code>	First description label
<code>\refitem{myList2{2}{first description label}}[\ref*, \label]</code>	2, First description label
<code>\refitem{myList2{2}{first description label}}[\ref\ \ref]</code>	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:

<code>\refitem{Important subitem}</code>	b)
<code>\refitem{Important subitem}[\label\ \label]</code>	2 – b)
<code>\refitem{Important subitem}[\ref* sub point \ref]</code>	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-hypbmsec` with package `hypbmsec`:
`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 `refname` 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*</code> <code>\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 <code>enumerate</code> 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 <code>CaSeS</code> <code>sEnSiTiV</code> 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 .log 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⟩]

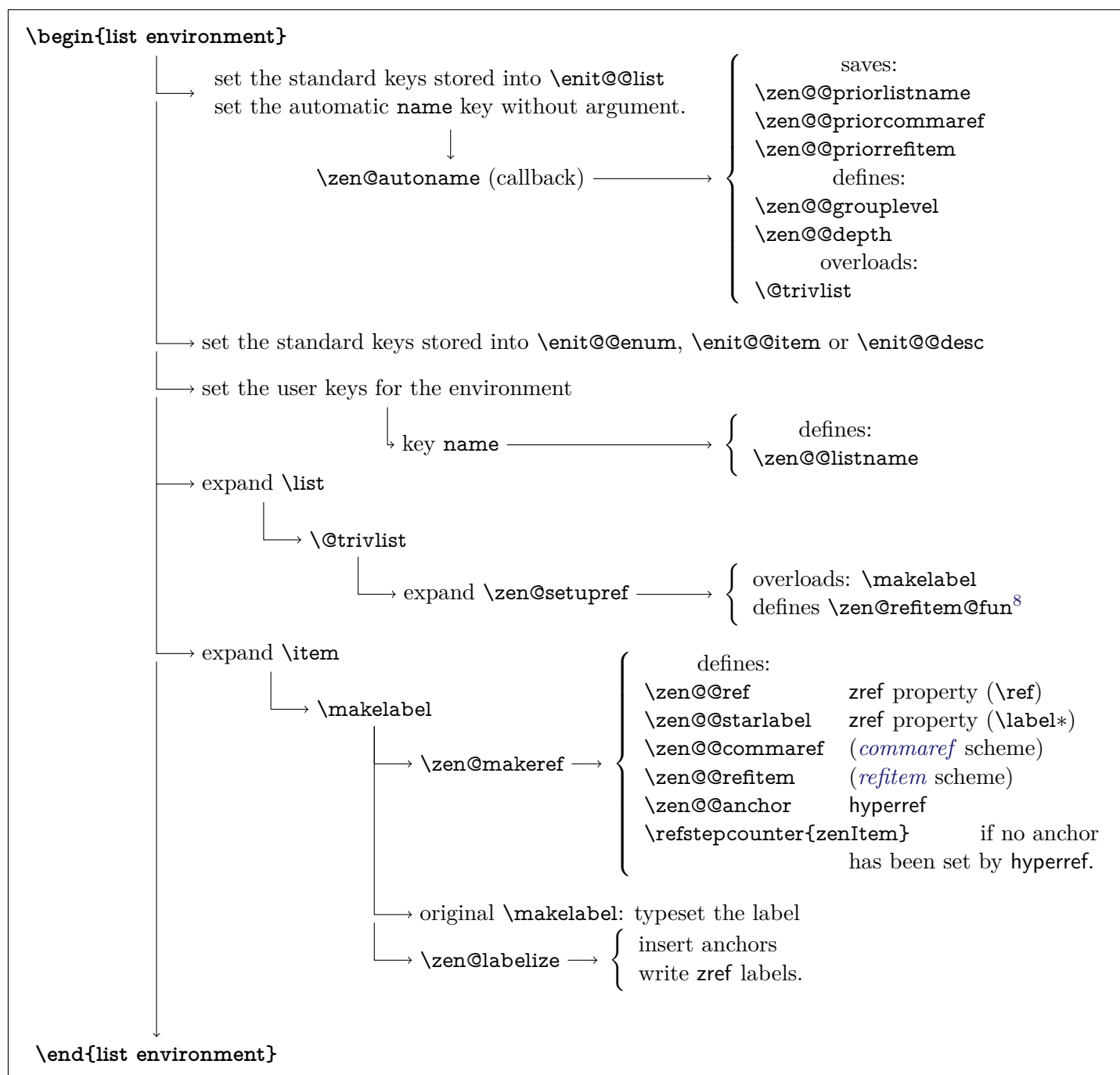
In the paragraph following an item, `\zeninfo` reports some informations about the references to the current item:

- 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.

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

- `enumitem-zref` is intended to use with L^AT_EX so we don't check if it is loaded twice.
- The package namespace is `\zen@`

```

1 \<*package>
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{enumitem-zref}[2010/12/17 v1.5 - 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{{%
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} % I (default comment: itemize option)
18 \TMP@EnsureCode{'\D}{14} % D (default comment: description option)
19 \TMP@EnsureCode{'\E}{9} % E (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@GreekORI \@Greek
40 \let\@greek \relax \let\@Greek \relax
41 \newalphalph\@greek[wrap]\zen@greekORI{24}%
42 \newalphalph\@Greek[wrap]\zen@GreekORI{24}%
43 \AddEnumerateCounter\greek\@greek{\@greek{23}}% widest is \psi
44 \AddEnumerateCounter\Greek\@Greek{\@Greek{12}}% widest is M
45 \let\zen@option@greekctr \relax
46 }% \zen@option@greekctr
47 \DeclareVoidOption{external}{\catcode'\E 14\relax}% E = 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@@norefreference` How to display undefined references.

```
72 \def\zen@@norefreference{\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@ \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@@noreference]{#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@@noreference]\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}\zen@@itemlabel
106 \zen@temp U{currentlabel}\zen@@currentlabel
107 \zen@temp U{currentlabelname}\zen@@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{\detokenize\expandafter{\GetTitleStringResult}}%
130     \def\noexpand\zen@temp {\detokenize\expandafter{\the\toks@}}%
131   }\x
132 }% \zen@Normalize
133 \def\zen@trailingspaces#1#2{\toks@{#1}\expandafter\zen@trailingsp@ces}
134 {\catcode47 = 8 % /
135 \gdef\zen@trailingsp@ces#1{%
136   \ifx \@nnil#1\@empty \expandafter\remove@to@nnil
137   \else \zen@{\toks@\expandafter{\the\expandafter\toks@
138     \expandafter{\romannumeral0\zen@postspace #1/ /}}}%

```

```

139 \expandafter\zen@trailingsp@ces % (loop)
140 \fi
141 }% \zen@trailingsp@ces
142 \gdef\zen@postspace#1 /{\zen@postsp@ce#1/}
143 \gdef\zen@postsp@ce#1/#2{ #1}
144 }% \catcode group
145 \def\zen@NormalizeCommands{%
146 \let\ensuremath \@firstofone % \ensuremath (engrec package)
147 \let\relax \@empty
148 \csname @safe@activetrue\endcsname
149 }% \zen@NormalizeCommands

```

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

```

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

```

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

```

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

```

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

```

168 \def\zen@starnamelist #1{% (1st automatic assignment to name done before)
169 \zen@Normalize\zen@CaSeS\zen@@listname{#1}%
170 \let\zen@@listname \zen@temp
171 \ifx \zen@@priorlistname\zen@@listname
172 \let\zen@@listname \@empty \fi
173 \zen@namelist\zen@@listname
174 }% \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.

```

175 \def\zen@autoname{% (1st automatic assignment to key name (by \enit@@list))
176 \unless \ifx\zen@@listname\@empty
177 \let\zen@@priorlistname \zen@@listname \fi
178 \unless \ifdefined\zen@@priorlistname \let\zen@@priorlistname \@empty \fi
179 \let\zen@@listname \@empty
180 \ifdefined\zen@@commaref \let\zen@@priorcommaref \zen@@commaref
181 \else
182 \let\zen@@commaref \@empty
183 \fi

```

```

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

```

210 \def\zen@trivlist{%
211 \let\@trivlist \zen@trivlistORI % restore original immediately
212 \@trivlist \zen@setuprefs
213 }% \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

```

214 \def\zen@type{%
215 \ifdefined\enit@type
216 \expandafter\strip@prefix \if e\enit@type>\let\zen@@type\z@ % enum begins with e
217 \else>\expandafter\strip@prefix \if i\enit@type>\let\zen@@type\@ne % item begins with i
218 \else>\expandafter\strip@prefix \if d\enit@type>\let\zen@@type\tw@ % desc begins with d
219 \else>\let\zen@@type\m@ne
220 \fi\fi\fi
221 \else \let\zen@@type\m@ne
222 \fi
223 }% \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`

```

224 \def\zen@setuprefs{%
225 \edef\zen@temp{\zen@@priorlistname\zen@@listname}%
226 \ifx \zen@temp\@empty
227 \ifx \makelabel\zen@makelabel \let\makelabel \zen@makelabelORI \fi
228 \else
229 \ifx \makelabel\zen@makelabel \else

```

```

230      \let \zen@makelabelORI \makelabel
231      \let \makelabel \zen@makelabel \fi
232 \fi
233 \let\zeninfo \zen@info
234 \ifcase \zen@type % enumerate only (key start)
235   \ifnum \c@zenidx=\z@ \c@zenidx\value\@enumctr\relax \fi
236 \fi
237 \zen@refitem@implicit
238 }% \zen@setuprefs
239 \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`).

```

240 \def\zen@refitem@implicit{% determine value of key refitem according to the label
241   \def\zen@gtemp {\the\c@zenidx}}%
242   \ifnum \zen@type=\z@ % enumerate
243     {\let\enit@refstar \zen@refitemfromlabel
244      \enit@labellist \zen@box{\@itemlabel}}}%
245   \fi
246   \let\zen@refitem@fun \zen@gtemp
247 }% \zen@refitem@implicit
248 \def\zen@refitemfromlabel#1{\def#1##1{0%
249   \def\zen@temp{##1}\ifx \zen@temp\@enumctr
250     \gdef\zen@gtemp{{#1{##1}}}\fi}%
251 }% \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.

```

252 \def\zen@makeref#1{% #1 = argument of \makelabel
253   \def\zen@@itemlabel{#1}\advance\c@zenidx \@ne
254   \ifx \zen@@itemlabel\zen@@autolabel
255     \if@nmbrlist
256       \global\advance\c@zenItem \@ne
257       \let\zen@@ref \@currentlabel
258     \else
259       {\refstepcounter{zenItem}}%
260       \let\zen@@ref \zen@refitem@fun
261     \fi
262     \let\zen@@starlabel \zen@refitem@fun
263     \zen@Normalize\zen@lowercase\zen@currentrefitem \zen@refitem@fun
264   \else % user defined label
265     {\refstepcounter{zenItem}}%
266     \zen@Normalize\zen@lowercase\zen@@ref {\zen@@itemlabel}}%
267     \let\zen@@starlabel \zen@@ref
268   \fi
269   \let\zen@currentrefitem \zen@temp % \zen@temp defined by \zen@Normalize
270   \let\zen@@anchor \@currentHref
271   \zen@makeref@item@comma
272 }% \zen@makeref

```

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

`zlabels` are written into the `.aux` file.

```

273 \def\zen@labelize{%
274   \ifzen@Hy \zen@anchors
275     \zref@labelbyprops{zen->\@currentHref}{zen@}\fi
276   \zref@labelbylist\zen@{zen@list}%
277   \zen@keeplabel
278 }% \zen@makeref

```


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

```

279 \def\zen@makeref@item@comma{%
280 \begingroup % to avoid silent assignment to \relax
281 \edef\x{\endgroup
282 \def\noexpand\zen@@commaref{%
283 \expandafter\zen@commaref@\zen@@priorcommaref \@nnil}%
284 \def\noexpand\zen@@refitem{%
285 \expandafter\zen@refitem@\zen@@priorrefitem \@nnil}%
286 }\x
287 }% \zen@makeref@item@comma
288 \def\zen@refitem@#1{%
289 \ifx \@nnil#1%
290 \ifx \zen@@listname\@empty \else
291 {\zen@@listname\zen@currentrefitem}\fi
292 \else {#1\zen@currentrefitem}\expandafter\zen@refitem@ % loop
293 \fi
294 }% \zen@refitem@
295 \def\zen@commaref@#1{%
296 \ifx \@nnil#1%
297 \ifx \zen@@listname\@empty \else
298 {\zen@@listname,\the\c@zenidx}\fi
299 \else {#1,\the\c@zenidx}\expandafter\zen@commaref@ % loop
300 \fi
301 }% \zen@commaref@

```

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

```

302 \let\zen@list@anchors\@empty
303 \def\zen@anchors{\begingroup
304 \def\zen@@anchors{\endgroup}%
305 \expandafter\expandafter\expandafter\zen@anchors@
306 \expandafter\zen@@commaref \zen@@refitem \@nnil
307 \zen@@anchors
308 }% \zen@anchors
309 \def\zen@anchors@#1{%
310 \ifx \@nnil#1\else
311 \expandafter\in@\c@name refitem.#1\expandafter\endcsname\expandafter{\zen@list@anchors}%
312 \ifin@ \zen@anchor@ignore{#1}%
313 \else
314 \edef\zen@@anchors{\zen@@anchors\zen@Hy@anchor{refitem.#1}}%
315 \xdef\zen@list@anchors{%
316 \unexpanded\expandafter{\zen@list@anchors}%
317 \expandafter\noexpand\c@name refitem.#1\endcsname}%
318 \fi
319 \expandafter\zen@anchors@ % loop
320 \fi
321 }% \zen@anchors@ (loop)
322 \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 `\z@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.

```

323 \def\zen@keep#1{\def\noexpand#1{\unexpanded\expandafter{#1}}}%
324 \def\zen@keeplabel{%
325 \xdef\zen@gtemp{%
326 \noexpand\ifnum \zen@@grouplevel<\currentgrouplevel
327 \aftergroup \noexpand\zen@gtemp

```

```

328     \noexpand\else
329     \c@zenidx \the\c@zenidx\relax
330     \zen@keep \zen@refitem
331     \zen@keep \zen@commaref
332     \ifzen@Hy \zen@keep\zen@@anchor\fi
333     \noexpand\fi
334   }\aftergroup\zen@gtemp
335 }% \zen@keeplabel

```

4.9 Extracting zref properties

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

```

336 \newcommand*\zen@extract[3]{% #1=ref #2=prop #3=def
337   \csname zen@extract@#2\endcsname\expandafter\expandafter%
338   \expandafter{\zref@extractdefault{zen>\number#1}{zen@#2}{#3}}%
339 }% \zen@extract

```

4.10 Referencing the items: `\refitem`

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

```

340 \newrobustcmd*\refitem{\begingroup \ifstar
341   {\zen@refitemOpt \zen@refText }%      no hyperlink
342   {\zen@refitemOpt \zen@refHy }%
343 }% \refitem
344 \def\zen@refHy{%
345   %% \edef\zen@@anchor{\zen@extract\c@zenItem{anchor}}{\unexpanded}%
346   %% \ifx \zen@@anchor\@empty \zen@@text
347   %% \else
348   \hyperref{}{}{\zref@extract{zen>\the\c@zenItem}{zen@anchor}}\zen@@text
349   %% \fi
350 }% \zen@ref
351 \def\zen@refText{\zen@@text}

```

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

```

352 \def\zen@refitemOpt#1{\let\zen@ref@#1\@ifnextchar[\zen@refitemExternal \zen@refitemRef }
353 \def\zen@refitemRef #1{\@tempswatrue \let\zen@lastwarn \m@ne
354   \zen@Normalize\zen@lowercase\zen@@label {#1}%
355   \zen@ifrefundefined \zen@@label
356   {\expandafter\expandafter\expandafter \zen@refused
357     \csname\ifcsname zen@refitem(\zen@temp)\endcsname
358       zen@refitem(\zen@temp)\else
359       zen@norefitem\fi\endcsname>}%
360   \zen@refitemFromAnchor
361 }% \zen@refitemRef
362 \def\zen@norefitem {\z@>0}

```

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

```

363 \def\zen@refused#1>#2>{%
364   \ifcase#1\relax
365     \@tempswafalse \refused\zen@@label
366     \let\zen@ref@ \zen@refText % do not create a hyperlink if no reference
367   \or \let\zen@@label \zen@temp \c@zenItem =#2\relax
368   \else \let\zen@@label \zen@temp \zen@MultipleLabels
369     \c@zenItem =#2\relax
370   \fi
371   \zen@refitem@prop
372 }% \zen@refused

```

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

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

```

373 \def\zen@refitemFromAnchor{%
374   \edef\zen@@anchor{\getrefbykeydefault\zen@@label{anchor}}}%

```



```

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

392 \def\zen@warn#1{\let\zen@lastwarn#1%
393 \message{Package enumitem-zref Warning:
394 \ifcase#1 %
395 \string\refitem{\zen@@label}:
396 User references to \string\label\space^^J\zen@spaces
397 require hyperref and the appropriate options for enumitem-zref^^J\zen@spaces
398 \ifzen@Hy Check the options given to package enumitem-zref.
399 \else Package hyperref is not loaded.\fi
400 \or User reference does not point to an item!^^J\zen@spaces
401 \string\refitem{\zen@@label}
402 \fi\on@line^^J^^J}}
403 \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).

```

404 \def\zen@refitem@prop{\futurelet\zen@temp \zen@refitemProp}
405 \def\zen@gobblespace#1 {#1}%
406 \let\zen@space \relax
407 \def\zen@refitemProp{%
408 \ifcase 0\ifx \zen@temp\else
409   \ifx \zen@temp\@sptoken 1\else
410   2\fi\fi\relax
411   \expandafter \zen@refitemPrint
412 \or \let\zen@space\space
413   \expandafter\zen@gobblespace\expandafter\zen@refitem@prop
414 \else \expandafter\zen@refitemPrint \expandafter[\expandafter]%
415 \fi
416 }% \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`.

```

417 \long\def\zen@refitemPrint[#1]{\def\zen@@text{#1}%
418 \if@tempswa % got a reference
419   \ifx \zen@@text\@empty
420     \edef\zen@@text{\zref@extract\c@zenItem{default}\zen@@noreference}\zen@space}%
421   \else\zen@refitemText
422   \fi
423 \else
424   \ifx \zen@@text\@empty \def\zen@@text{\zen@@noreference\zen@space}%\refused\zen@@label
425   \else \zen@refitem@noText
426   \fi
427 \fi
428 \zen@ref@
429 \endgroup
430 }% \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`.

```

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

```

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

```

480 \def\zen@refitemExternal[#1]#2{%
481   \toks@{#1}\zen@Normalize\zen@lowercase\zen@@label {#2}%
482   \edef\zen@ref@{\endgroup
483     \noexpand\hyperref{\the\toks@}{refitem}{\zen@@label}%
484   }\zen@ref@
485 }% \zen@Hy@external

```

4.11 Undefined references and duplicate labels management

`\zen@MultipleLabels`

```

486 \def\zen@MultipleLabels{%
487   \PackageWarning{enumitem-zref}
488     {'\zen@@label' on page \thepage \space multiple defined\MessageBreak
489     Only the first one will be referenced}% et mettre ici la liste des références correspondant aux alias
490 }% \zen@MultipleLabels

```

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

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

```

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

```

`\zeninfo` is executed when out of scope:

```

511 \newrobustcmd*\zeninfo[1][\begingroup
512   \ifdefined\zen@@type
513   \ifcase \zen@@type \zeninfo@err
514   \or \zeninfo@err
515   \or \zeninfo@err
516   \else\zeninfo@err[{in environment \@currenvir\MessageBreak
517     You should be in a list environment}]}%
518   \fi\endgroup
519 }% \zeninfo (out of scope)
520 \newcommand*\zeninfo@err[1][\{you should give a name to your \@currenvir\space list}]}{%
521   \PackageError{enumitem-zref}
522     {\string\zeninfo\space is not available\MessageBreak
523     #1}\@eha
524 }% \zeninfo@err

```

`\zen@whichinfo`

```

525 \def\zen@whichinfo#1{%
526   \expandafter\strip@prefix   \if d#1>\z@   % depth (begins with d)
527   \else>\expandafter\strip@prefix\if i#1>\@ne % index (begins with i)
528   \else>\expandafter\strip@prefix\if t#1>\tw@ % type (begins with t)
529   \else>\expandafter\strip@prefix\if a#1>\thr@@ % anchor (begins with a)
530   \else>\expandafter\strip@prefix\if r#1>4      % refitem(begins with r)
531   \else>\expandafter\strip@prefix\if c#1>5      % commaref(begins with c)
532   \else>\m@ne
533   \fi\fi\fi\fi\fi
534 }% \zen@whichinfo

```

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

`\zen@item@comma`

```

535 \def\zen@item@comma{%
536   \begingroup \c@zenidx \@ne
537   \@whilsw \ifcsname Z@R@zen>\the\c@zenidx\endcsname\fi
538   {\advance\c@zenidx \@ne}%
539   \@whilenum \c@zenidx>\@ne
540   \do{\advance \c@zenidx \m@ne
541     \zref@wrapper@unexpanded{%
542       \edef\zen@refitem{%
543         \zref@extractdefault{zen>\the\c@zenidx}{zen@refitem}{}}%
544         \zref@extractdefault{zen>\the\c@zenidx}{zen@commaref}{}}}%
545     \edef\zen@refitem{\expandafter
546       \zen@item@comma@ \zen@refitem \@nnil}%
547     \expandafter\zen@def@item@comma@ \zen@refitem \@nnil}}%
548   \endgroup
549 }% \zen@item@comma
550 \def\zen@item@comma@ #1{%
551   \ifx\@nnil#1\else
552     \expandafter\noexpand\csname zen@refitem(\detokenize{#1})\endcsname
553     \expandafter\zen@item@comma@ % (loop)
554   \fi
555 }% \zen@item@comma@

```

`\zen@def@item@comma@`

```

556 \def\zen@def@item@comma@ #1{\ifx\@nnil#1\else
557   \zen@def@item@comma@{#1}\expandafter\zen@def@item@comma@ \fi % loop
558 }% \zen@def@item@comma@

```

`\zen@def@item@comma`

```

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

578 \zen@AtEnd \let\zen@AtEnd\@undefined
579 \</package>

```

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

[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`⁹ package while `greek` is based on `engrec`¹⁰. Both uses code of `alphalph`¹¹ 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.

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		
<code>\@Grec</code>	28, 29, 31, 33	<code>\@trivlist</code> 195, 196, 211, 212
<code>\@Greek</code>	39, 40, 42, 44	<code>\@whilenum</code> 539
<code>\@currentHref</code>	270, 275	<code>\@whilesw</code> 537
<code>\@currentlabel</code>	106, 257	
<code>\@currentlabelname</code>	107	A
<code>\@currenvir</code>	205, 516, 520	<code>\AddEnumerateCounter</code> 32, 33, 43, 44
<code>\@enumctr</code>	235, 249	<code>\aftergroup</code> 327, 334
<code>\@grec</code>	28, 29, 30, 32	<code>\anchor</code> 444, 476
<code>\@greek</code>	39, 40, 41, 43	<code>\AtEndOfPackage</code> 25, 36
<code>\@ifpackageloaded</code>	50, 51	<code>\AtEndPreamble</code> 67
<code>\@itemlabel</code>	71, 105, 244	
<code>\@listdepth</code>	98, 189	C
		<code>\c@zenidx</code> 68, 99, 194, 201, 208, 235, 241, 253, 298, 299, 329, 432, 433, 456,
9. <code>greekctr</code> : CTAN:help/Catalogue/entries/greekctr.html		
10. <code>engrec</code> : CTAN:help/Catalogue/entries/engrec.html		
11. <code>alphalph</code> : CTAN:help/Catalogue/entries/alphalph.html		

457, 458, 462, 467, 469, 470, 472, 494, 503, 536, 537, 538, 539, 540, 543, 544, 563, 565, 571	
\c@zenItem	84, 190,
256, 345, 348, 367, 369, 387, 420, 434, 435, 458	
\commaref	444, 477
\count@	10, 12
\counter	440, 475
\currentgrouplevel	155, 188, 326
\currentlabel	445, 477
\currentlabelname	446, 477
D	
\DeclareBoolOption	21, 22, 23, 24
\DeclareDefaultOption	48
\DeclareVoidOption	25, 36, 47
\default	440, 475
\define@key	150, 151
\depth	443, 476
\detokenize	89, 129, 130, 498, 499, 505, 506, 552
E	
\EnGrec	33
\engrec	32
\enit@list	152, 156, 175
\enit@afterlist	202, 203
\enit@endenumerate	198, 199
\enit@format	7
\enit@labellist	244
\enit@refstar	243
\enit@type	96, 215, 216, 217, 218
\ensuremath	146
F	
\frenchbsetup	67
\futurelet	404
G	
\G@refundefinedtrue	376, 381
\getrefbykeydefault	374
\GetTitleStringDisableCommands	7
\GetTitleStringExpand	125
\GetTitleStringResult	123, 124, 126, 129
\Greek	44
\greek	43
H	
\hbadness	81
\hfuzz	81
\Hy@raisedlink	74, 75
\hyper@anchor	117
\hyper@anchorend	74
\hyper@anchorstart	74
\hyperref	348, 483
I	
\if@nmbrlist	255
\ifcsname	357, 537
\ifin@	312
\ifzen@CaSeS	77
\ifzen@description	56, 65
\ifzen@duplicatealiases	559, 573
\ifzen@enumerate	54, 59
\ifzen@Hy	115, 274, 332, 382, 398, 497, 507
\ifzen@itemize	55, 64
\in@	311
\index	442, 476
L	
\label	396, 439, 451, 474
\listctr	442, 476
\listname	441, 475
\lowercase	78
\ltx@car	466
\ltx@GobbleNum	466
\ltx@ifUndefined	80
M	
\makelabel	227, 229, 230, 231, 252
\message	393
N	
\newalphalph	30, 31, 41, 42
\newcounter	68
\newtoks	70
\nfss@text	72
\none	475
\number	189, 338, 568
\numexpr	568
O	
\on@line	402
P	
\PackageError	521
\PackageInfo	322
\PackageWarning	60, 487
\page	445, 474
\ProcessLocalKeyvalOptions	53
R	
\ref	382, 383, 385, 438, 450, 451, 470, 474, 475, 476, 477, 479
\refitem	5, 6, 340, 395, 401, 443, 477
\refstepcounter	259, 265
\refused	365, 424
\remove@to@nnil	136
\romannumeral	138, 466
S	
\setcounter	66
\strip@prefix	216, 217, 218, 387, 526, 527, 528, 529, 530, 531, 565, 574, 575
T	
\thepage	92, 488
\tracinglostchars	82
\type	441, 476
V	
\value	100, 235, 504
\vbadness	81
\vbox	82
\vfuzz	81
Z	
\zen@	84, 85, 122, 137, 276, 452, 458, 459, 461, 464
\zen@@anchor	94, 270, 332, 345, 346, 374, 375, 379, 497, 507
\zen@@anchors	304, 307, 314
\zen@@autolabel	71, 254
\zen@@commaref	109, 180, 182, 282, 306, 331, 499, 505
\zen@@depth	189, 190, 435, 436, 456, 493, 502
\zen@@grouplevel	154, 155, 188, 326
\zen@@itemlabel	102, 253, 254, 266

<code>\zen@@label</code>	354, 355, 365, 367, 368, 374, 382, 383, 385, 395, 401, 424, 481, 483, 488
<code>\zen@@listname</code> . . .	97, 164, 165, 169, 170, 171, 172, 173, 176, 177, 179, 225, 290, 291, 297, 298
<code>\zen@noreference</code> . .	<u>72</u> , 87, 92, 420, 424, 461, 479
<code>\zen@parents</code>	101, 190, 191, 434, 467
<code>\zen@priorcommaref</code>	180, 181, 283
<code>\zen@priorlistname</code>	171, 177, 178, 225
<code>\zen@priorrefitem</code>	184, 185, 285
<code>\zen@ref</code>	95, 104, 257, 260, 266, 267
<code>\zen@refitem</code> 108, 184, 186, 284, 306, 330, 498, 506	
<code>\zen@starlabel</code>	103, 262, 267
<code>\zen@text</code> 346, 348, 351, 417, 419, 420, 424, 447, 471	
<code>\zen@type</code>	96, 157, 193, 216, 217, 218, 219, 221, 234, 242, 496, 504, 512, 513
<code>\zen@ancestor</code>	459, 466
<code>\zen@ancestorprop</code>	438, 439, 440, 441, 442, 443, 444, 445, 446, 455
<code>\zen@ancestorproperty</code>	465
<code>\zen@anchor@ignore</code>	312, 322
<code>\zen@anchors</code>	274, <u>302</u>
<code>\zen@anchors@</code>	305, 309, 319, 321
<code>\zen@autoname</code>	158, 160, 162, <u>175</u>
<code>\zen@box</code>	<u>81</u> , 244, 447, 471
<code>\zen@CaSeS</code>	<u>76</u> , 164, 169
<code>\zen@clprops</code>	450, 470, 474, 478, 479
<code>\zen@commaref@</code>	283, 295, 299, 301
<code>\zen@currentrefitem</code>	263, 269, 291, 292
<code>\zen@def@item@comma</code>	557, <u>559</u>
<code>\zen@def@item@comma@</code>	547, <u>556</u>
<code>\zen@descriptiontrue</code>	57
<code>\zen@duplicatealiasestrue</code>	575
<code>\zen@enumerate</code>	199, 201
<code>\zen@enumeratetrue</code>	55, 56, 57, 60
<code>\zen@extract</code>	<u>336</u> , 345, 420, 434, 435, 461
<code>\zen@getancestor</code>	468
<code>\zen@gobblespace</code>	405, 413
<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>	241, 246, 250, 325, 327, 334
<code>\zen@Hy@anchor</code>	<u>73</u> , 314
<code>\zen@Hy@external</code>	485
<code>\zen@Hyfalse</code>	119
<code>\zen@Hytrue</code>	117
<code>\zen@ifrefundefined</code>	80, 355
<code>\zen@info</code>	233, <u>491</u>
<code>\zen@item@comma</code>	66, <u>535</u>
<code>\zen@item@comma@</code>	546, 550, 553, 555
<code>\zen@itemizetrue</code>	57
<code>\zen@keep</code>	323, 330, 331, 332
<code>\zen@keeplabel</code>	277, <u>323</u>
<code>\zen@labelize</code>	239, <u>273</u>
<code>\zen@lastwarn</code>	353, 392
<code>\zen@list@anchors</code>	302, 311, 315, 316
<code>\zen@lowercase</code>	<u>76</u> , 263, 266, 354, 481
<code>\zen@makelabel</code>	<u>224</u>
<code>\zen@makelabelORI</code>	227, 230, 239
<code>\zen@makeref</code>	239, <u>252</u> , 278
<code>\zen@makeref@item@comma</code>	271, <u>279</u>
<code>\zen@MultipleLabels</code>	368, <u>486</u>
<code>\zen@name@list</code>	197
<code>\zen@namelist</code>	150, <u>152</u> , 173
<code>\zen@nil</code>	449, 452
<code>\zen@norefitem</code>	362
<code>\zen@Normalize</code> <u>122</u> , 164, 169, 263, 266, 269, 354, 481	
<code>\zen@NormalizeCommands</code>	125, 145, 149
<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>	142, 143
<code>\zen@postspace</code>	138, 142
<code>\zen@prepare@resume</code>	193, 198, 200, 209
<code>\zen@propcount</code>	431, 469, 473
<code>\zen@ref</code>	350
<code>\zen@ref@</code> 352, 366, 377, 382, 383, 385, 428, 482, 484	
<code>\zen@refHy</code>	119, 342, 344, 382
<code>\zen@refitem</code>	542, 545, 546, 547
<code>\zen@refitem@</code>	285, 288, 292, 294
<code>\zen@refitem@fun</code>	192, 246, 260, 262, 263
<code>\zen@refitem@implicit</code>	237, <u>240</u>
<code>\zen@refitem@noText</code>	425, 479
<code>\zen@refitem@prop</code>	371, 390, <u>404</u>
<code>\zen@refitemExternal</code>	352, <u>480</u>
<code>\zen@refitemFromAnchor</code>	360, <u>373</u>
<code>\zen@refitemfromlabel</code>	243, 248, 251
<code>\zen@refitemOpt</code>	341, 342, <u>352</u>
<code>\zen@refitemPrint</code>	411, 414, <u>417</u>
<code>\zen@refitemProp</code>	404, 407, 416
<code>\zen@refitemRef</code>	352, 353, 361
<code>\zen@refitemText</code>	421, <u>431</u>
<code>\zen@refText</code>	119, 351, 366, 377
<code>\zen@reftext</code>	341
<code>\zen@refused</code>	356, <u>363</u>
<code>\zen@setuprefs</code>	212, <u>224</u>
<code>\zen@space</code>	406, 412, 420, 424
<code>\zen@spaces</code>	396, 397, 400, 403
<code>\zen@starnamelist</code>	151, <u>168</u>
<code>\zen@temp</code> 86, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 130, 165, 170, 225, 226, 249, 269, 357, 358, 367, 368, 379, 380, 387, 404, 408, 409, 461, 462, 464	
<code>\zen@toks</code> <u>68</u> , 202, 203, 448, 449, 450, 451, 452, 464	
<code>\zen@trailingsp@ces</code>	133, 135, 139, 141
<code>\zen@trailingspaces</code>	126, 133
<code>\zen@trivlist</code>	196, <u>210</u>
<code>\zen@trivlistORI</code>	195, 211
<code>\zen@type</code>	156, <u>214</u>
<code>\zen@unique</code>	83
<code>\zen@warn</code>	376, 381, 392
<code>\zen@whichinfo</code>	493, <u>525</u>
<code>\zeninfo</code>	7, 233, <u>511</u>
<code>\zeninfo@err</code>	513, 514, 515, 516, 520, 524
<code>\zenItem</code>	68
<code>\zref@addprop</code>	111, 118
<code>\zref@extract</code>	348
<code>\zref@extractdefault</code>	338, 379, 543, 544
<code>\zref@getcurrent</code>	110
<code>\zref@labelbylist</code>	276
<code>\zref@labelbyprops</code>	275
<code>\zref@newlist</code>	83
<code>\zref@newprop</code>	85, 87, 92
<code>\zref@wrapper@unexpanded</code>	541
È	
<code>\È</code>	19, 47
Ĭ	
<code>\Ĭ</code>	17, 64
Đ	
<code>\Đ</code>	18, 65