

The `zref` package

Heiko Oberdiek*

2026-09-16 v2.38

Abstract

Package `zref` tries to get rid of the restriction in $\text{\LaTeX}$'s reference system that only two properties are supported. The package implements an extensible referencing system, where properties are handled in a more flexible way. It offers an interface for macro programmers for the access to the system and some applications that uses the new reference scheme.

Contents

1	Introduction	4
1.1	Standard $\text{\LaTeX}$ behaviour	4
1.2	History of the basic idea	5
1.3	Interfaces	5
2	Interface for programmers	5
2.1	Entities	5
2.2	Property list	6
2.3	Property	7
2.4	Reference generation	7
2.5	Data extraction	8
2.6	Setup	9
2.7	Declared properties	10
2.8	Wrapper for advanced situations	10
2.9	Counter for unique names	11
3	User interface	11
3.1	Module <code>user</code>	11
3.2	Module <code>abspage</code>	13
3.3	Module <code>lastpage</code>	13
3.3.1	Tests for last page	13
3.3.2	Example	14
3.4	Module <code>thepage</code>	14
3.5	Module <code>nextpage</code>	15
3.5.1	Configuration	15
3.5.2	Example	15
3.6	Module <code>totpages</code>	16
3.7	Module <code>pagelayout</code>	16
3.8	Module <code>marks</code>	17
3.9	Module <code>runs</code>	17
3.10	Module <code>perpage</code>	17
3.11	Module <code>counter</code>	18
3.12	Module <code>titleref</code>	18
3.13	Module <code>savepos</code>	19
3.14	Module <code>abspos</code>	20

*Please report any issues at <https://github.com/LaTeX-Package-Repositories/zref>

3.15	Module <code>dotfill</code>	20
3.16	Module <code>env</code>	21
3.17	Module <code>xr</code>	21
3.18	Module <code>pageattr</code>	21
4	ToDo	22
5	Example	22
6	Implementation	24
6.1	Package <code>zref</code>	24
6.1.1	Identification	24
6.1.2	Load basic module	24
6.1.3	Process options	25
6.2	Module <code>base</code>	25
6.2.1	Prefixes	25
6.2.2	Identification	26
6.2.3	Utilities	26
6.2.4	Check for ε -TeX	27
6.2.5	Auxiliary file stuff	27
6.2.6	Property lists	28
6.2.7	Properties	32
6.2.8	Reference generation	35
6.2.9	Reference querying and extracting	38
6.2.10	Compatibility with <code>babel</code>	42
6.2.11	Unique counter support	42
6.2.12	Utilities	43
6.2.13	Setup	43
6.3	Module <code>user</code>	44
6.4	Module <code>abspage</code>	46
6.5	Module <code>counter</code>	46
6.6	Module <code>lastpage</code>	47
6.7	Module <code>thepage</code>	48
6.8	Module <code>nextpage</code>	49
6.9	Module <code>totpages</code>	50
6.10	Module <code>pagelayout</code>	51
6.10.1	Define layout properties	51
6.11	Module <code>pageattr</code>	54
6.12	Module <code>marks</code>	58
6.13	Module <code>runs</code>	59
6.14	Module <code>perpage</code>	60
6.15	Module <code>titleref</code>	62
6.15.1	Implementation	63
6.15.2	User interface	64
6.15.3	Patches for section and caption commands	65
6.15.4	Environment <code>description</code>	66
6.15.5	Class <code>memoir</code>	66
6.15.6	Class <code>beamer</code>	67
6.15.7	Package <code>titlesec</code>	67
6.15.8	Package <code>longtable</code>	67
6.15.9	Package <code>listings</code>	68
6.15.10	Theorems	68
6.16	Module <code>xr</code>	68
6.17	Module <code>hyperref</code>	76
6.18	Module <code>savepos</code>	76
6.18.1	Identification	77
6.18.2	Availability	77
6.18.3	Setup	77

6.18.4	User macros	77
6.19	Module abspos	78
6.19.1	Identification	78
6.19.2	Media	81
6.19.3	Paper	83
6.19.4	Origin	83
6.19.5	Header	84
6.19.6	Body	85
6.19.7	Footer	86
6.19.8	Marginal notes	86
6.19.9	Stock paper	87
6.20	Module dotfill	88
6.21	Module env	89
7	Installation	89
7.1	Download	89
7.2	Package installation	90
7.3	Refresh file name databases	90
7.4	Some details for the interested	90
8	References	91
9	History	91
	[2006/02/20 v1.0]	91
	[2006/05/03 v1.1]	91
	[2006/05/25 v1.2]	91
	[2006/09/08 v1.3]	91
	[2007/01/23 v1.4]	91
	[2007/02/18 v1.5]	92
	[2007/04/06 v1.6]	92
	[2007/04/17 v1.7]	92
	[2007/04/22 v1.8]	92
	[2007/05/02 v1.9]	92
	[2007/05/06 v2.0]	92
	[2007/05/28 v2.1]	92
	[2008/09/21 v2.2]	92
	[2008/10/01 v2.3]	92
	[2009/08/07 v2.4]	92
	[2009/12/06 v2.5]	92
	[2009/12/07 v2.6]	93
	[2009/12/08 v2.7]	93
	[2010/03/26 v2.8]	93
	[2010/03/29 v2.9]	93
	[2010/04/08 v2.10]	93
	[2010/04/15 v2.11]	93
	[2010/04/17 v2.12]	93
	[2010/04/19 v2.13]	93
	[2010/04/22 v2.14]	94
	[2010/04/23 v2.15]	94
	[2010/04/28 v2.16]	94
	[2010/05/01 v2.17]	94
	[2010/05/13 v2.18]	94
	[2010/10/22 v2.19]	94
	[2011/02/12 v2.20]	94
	[2011/03/18 v2.21]	95
	[2011/10/05 v2.22]	95
	[2011/12/05 v2.23]	95
	[2012/04/04 v2.24]	95

[2016/05/16 v2.25]	95
[2016/05/21 v2.26]	95
[2018/11/21 v2.27]	95
[2019/11/29 v2.28]	95
[2020-03-03 v2.29]	95
[2020-03-04 v2.30]	96
[2020-05-28 v2.31]	96
[2020-07-03 v2.32]	96
[2022-03-08 v2.33]	96
[2022-04-07 v2.34]	96
[2023-09-14 v.2.35]	96
[2025-06-08 v2.36]	96
[2026-03-07 v2.37]	96
[2026-09-16 v2.38]	96

10 Index	97
-----------------	-----------

1 Introduction

Standard L^AT_EX's reference system with `\label`, `\ref`, and `\pageref` now supports five properties, the appearance of the counter that is last incremented by `\refstepcounter` (stored in `\@currentlabel`), the page, a title (if it has been stored in `\@currentlabelname`), the current anchor (stored in `\@currentHref`) and an external file name if the reference is imported by the `xr` package with the `\label` command.

Unhappily for a long time L^AT_EX did provide only two properties (the counter and the page) and did not provide an interface for adding another properties. Packages such as `hyperref`, `nameref`, or `titleref` were forced to use ugly hacks to extend the reference system. These ugly hacks were one of the causes for `hyperref`'s difficulty regarding compatibility with other packages.

1.1 Standard L^AT_EX behaviour

References are created by the `\label` command:

```
\chapter{Second chapter}
\section{First section on page 7} % section 2.1
\label{myref}
```

Now L^AT_EX records the section number 2.1, the page 7, a title and an anchor (if, e.g., `hyperref` has been loaded), in the reference. Internally the reference is a list with five entries:

```
\r@myref → {2.1}{7}{First section on page 7}{section.2.1}{}
```

The length of the list is fixed in the L^AT_EX kernel, An interface for adding new properties is missing.

There were several tries to add new properties:

hyperref used a list of five properties instead of the standard list with two entries. This caused many compatibility problems with L^AT_EX and other packages, but as the L^AT_EX kernel extended the standard list to five entries too, this problem have been resolved.

titleref stores its title data into the first entry in the list. L^AT_EX was happy because it did only see its standard list with two entries, but solution was also not easy to extend to more fields: Then the macros form a nested structure inside the first reference argument for the label. Expandable extractions then becomes painful. `titleref` is not compatible with `hyperref` and since L^AT_EX extended the number of fields it is not longer needed.

1.2 History of the basic idea

Some time ago Morten Høgholm sent me an experimental cross referencing mechanism as “expl3” code. His idea is:

```
\g_xref_mylabel_plist →  
  \xref_dance_key{salsa}\xref_name_key{Morten}...
```

The entries have the following format:

```
\xref_⟨your key⟩_key{⟨some text⟩}
```

This approach is much more flexible:

- New properties can easily be added, just use a new key.
- The length of the list is not fixed. A reference can use a subset of the keys.
- The order of the entries does not matter.

Unhappily I am not familiar with the experimental code for L^AT_EX3 that will need some time before its first release. Thus I have implemented it as L^AT_EX 2_ε package without disturbing the existing L^AT_EX reference system.

Note that a similar idea is now implemented in L^AT_EX directly in `ltproperties`.

1.3 Interfaces

The package provides a generic *interface for programmers*. Commands of this interface are prefixed by `\zref@`.

Option `user` enables the *user interface*. Here the commands are prefixed by `\z` to avoid name clashes with existing macros.

Then the packages provides some *modules*. They are applications for the reference system and can also be considered as examples how to use the reference system.

The modules can be loaded as packages. The package name is prefixed with `zref-`, for example:

```
RequirePackage{zref-abspage}
```

This is the preferred way if the package is loaded from within other packages to avoid option clashes.

As alternative package `zref` can be used and the modules are given as options:

```
usepackage[perpage,user]{zref}
```

2 Interface for programmers

The user interface is described in the next section [3](#).

2.1 Entities

Reference. Internally a reference is a list of key value pairs:

```
\Z@R@myref → \default{2.1}\page{7}
```

The generic format of a entry is:

```
\Z@R@⟨refname⟩ → \⟨propname⟩{⟨value⟩}
```

⟨*refname*⟩ is the name that denoted references (the name used in `\label` and `\ref`).
⟨*propname*⟩ is the name of the property or key. The property key macro is never executed, it is used in parameter text matching only.

Property. Because the name of a property is used in a macro name that must survive the .aux file, the name is restricted to letters and ‘@’.

Property list. Often references are used for special purposes. Thus it saves memory if just the properties are used in this reference that are necessary for its purpose.

Therefore this package uses the concept of *property lists*. A property list is a set of properties. The set of properties that is used by the default \label command is the *main property list*.

2.2 Property list

^{exp} means that the implementation of the marked macro is expandable. ^{exp2} goes a step further and marks the macro expandable in exact two expansion steps.

```
\zref@newlist {\listname}
```

Declares a new empty property list.

```
\zref@addprop {\listname} {\propname}
\zref@localaddprop {\listname} {\propname}
```

Adds the property *\propname* to the property list *\listname*. The property and list must exist. The addition is global by \zref@addprop and limited to local scope by \zref@localaddprop. Between 2010/04/19 v2.13 and 2010/10/22 v2.19 a comma separated list of properties could be used as argument *\propname*. Since 2010/10/22 v2.19 the addition of several properties at once is supported by \zref@addprops.

```
\zref@addprops {\listname} {\propname list}
\zref@localaddprops {\listname} {\propname list}
```

These macros add a comma separated list of properties *\propname list* to list *\listname*. \zref@addprops works globally and \zref@localaddprops locally. Since 2010/10/22 v2.19.

```
\zref@delprop {\listname} {\propname}
\zref@localdelprop {\listname} {\propname}
```

Removes the property *\propname* from the property list *\listname*. The property and list must exist. The removal is global by \zref@delprop and limited to local scope by \zref@localdelprop.

```
\zref@showprop {\propname}
\zref@showprop {\propname}
```

Writes (only) into the log-file some information about the propname.

```
\zref@listexists {\listname} {\then}
```

Executes *\then* if the property list *\listname* exists or raise an error otherwise.

```
\zref@iflistundefinedexp {\listname} {\then} {\else}
```

Executes *\then* if the list exists or *\else* otherwise.

`\zref@iflistcontainsprop {<listname>} {<propname>} {<then>} {<else>}`

Executes `<then>` if the property `<propname>` is part of property list `<listname>` or otherwise it runs the `<else>` part.

2.3 Property

`\zref@newprop* {<propname>} [ <default> ] {<value>}`

This command declares and configures a new property with name `<propname>`.

In case of unknown references or the property does not exist in the reference, the `<default>` is used as value. If it is not specified here, a global default is used, see `\zref@setdefault`.

The correct values of some properties are not known immediately but at page shipout time. Prominent example is the page number. These properties are declared with the star form of the command.

`\zref@setcurrent {<propname>} {<value>}`

This sets the current value of the property `<propname>`. It is a generalization of setting L^AT_EX's `\currentlabel`.

`\zref@getcurrentexp2 {<propname>}`

This returns the current value of the property `<propname>`. The value may not be correct, especially if the property is bound to a page (start form of `\zref@newprop`) and the right value is only known at shipout time (e.g. property ‘page’). In case of errors (e.g. unknown property) the empty string is returned.

Since version 2010/04/22 v2.14 `\zref@getcurrent` supports `\zref@wrapper@unexpanded`.

`\zref@propexists {<propname>} {<then>}`

Calls `<then>` if the property `<propname>` is available or generates an error message otherwise.

`\zref@ifpropundefinedexp {<propname>} {<then>} {<else>}`

Calls `<then>` or `<else>` depending on the existence of property `<propname>`.

2.4 Reference generation

`\zref@label {<refname>}`

This works similar to `\label`. The reference `<refname>` is created and put into the `.aux` file with the properties of the main property list.

`\zref@labelbylist {<refname>} {<listname>}`

Same as `\zref@label` except that the properties are taken from the specified property list `<listname>`.

`\zref@labelbyprops {<refname>} {<propnameA>,<propnameB>,...}`

Same as `\zref@label` except that these properties are used that are given as comma separated list in the second argument.

`\zref@labelbykv {<key value list>} {<refname>}`

Same as `\zref@label` except that a key value list is used to select the properties. See the description of `\zkvlabel`, page 12.

`\zref@newlabel {<refname>} {...}`

This is the macro that is used in the `.aux` file. It is basically the same as `\newlabel` apart from the format of the data in the second argument.

2.5 Data extraction

`\zref@extractdefaultexp2 {<refname>} {<propname>} {<default>}`

This is the basic command that references the value of a property `<propname>` for the reference `<refname>`. In case of errors such as undefined reference the `<default>` is used instead.

`\zref@extractexp2 {<refname>} {<propname>}`

The command is an abbreviation for `\zref@extractdefault`. As default the default of the property is taken, otherwise the global default.

Example for page references:

```

LATEX: \pageref{foobar}
zref:   \zref@extract{foobar}{page}
```

Both `\zref@extract` and `\zref@extractdefault` are expandable. That means, these macros can directly be used in expandable calculations, see the example file. On the other side, babel's shorthands are not supported, there are no warnings in case of undefined references.

If an user interface doesn't need expandable macros then it can use `\zref@refused` and `\zref@wrapper@babel` for its user macros.

`\zref@refused {<refname>}`

This command is not expandable. It causes the warnings if the reference `<refname>` is not defined. Use the `\zref@extract` commands inside expandable contexts and mark their use outside by `\zref@refused`, see the example file.

`\zref@def@extract {<cmd>} {<refname>} {<propname>}`
`\zref@def@extractdefault {<cmd>} {<refname>} {<propname>} {<default>}`

Both macros extract the property `<propname>` from the reference `<refname>` the same way as macros `\zref@extract` and `\zref@extractdefault`. The result is stored in macro `<cmd>`. Also `\zref@refused` is called to notify L^AT_EX that the reference `<refname>` is used. Added in 2011/10/04 v2.22.

`\zref@ifrefundefinedexp {<refname>} {<then>} {<else>}`

Macro `\zref@ifrefundefined` calls arguments `<then>` or `<else>` dependent on the existence of the reference `<refname>`.

`\zifrefundefined {<refname>} {<then>} {<else>}`

Macro `\zifrefundefined` calls `\ref@refused` before executing `\zref@ifrefundefined`. Babel shorthands are supported in `<refname>`.

`\zref@ifrefcontainspropexp {<refname>} {<propname>} {<then>} {<else>}`

Test whether a reference provides a property.

2.6 Setup

`\zref@default`

Holds the global default for unknown values.

`\zref@setdefault {<value>}`

Sets the global default for unknown values. The global default is used, if a property does not specify an own default and the value for a property cannot be extracted. This can happen if the reference is unknown or the reference does not have the property.

`\zref@setmainlist {<value>}`

Sets the name of the main property list. The package sets and uses `main`.

2.7 Declared properties

Module	Property	Property list	Default
(base)	default	main	<code><empty></code>
	page	main	<code><empty></code>
abspage	abspage	main	0
counter	counter	main	<code><empty></code>
hyperref	anchor	main	<code><empty></code>
	url		<code><empty></code>
pageattr	pdfpageattr	thepage	...
	pdfpagesattr	LastPage	...
pagelayout ¹	mag	thepage	<code>\number\mag</code>
	paperwidth	thepage	<code>\number\paperwidth</code>
	paperheight	thepage	<code>\number\paperheight</code>
	stockwidth	thepage	<code>\number\stockwidth</code>
	stockheight	thepage	<code>\number\stockheight</code>
	pdfpageheight	thepage	<code>\number\pdfpageheight</code>
	pdfpagewidth	thepage	<code>\number\pdfpagewidth</code>
	pdfhorigin	thepage	<code>\number\pdfhorigin</code>
	pdfvorigin	thepage	<code>\number\pdfvorigin</code>
	hoffset	thepage	<code>\number\hoffset</code>
	voffset	thepage	<code>\number\voffset</code>
	topmargin	thepage	<code>\number\topmargin</code>
	oddsidemargin	thepage	<code>\number\oddsidemargin</code>
	evensidemargin	thepage	<code>\number\evensidemargin</code>
	textwidth	thepage	<code>\number\textwidth</code>
	textheight	thepage	<code>\number\textheight</code>
	headheight	thepage	<code>\number\headheight</code>
	headsep	thepage	<code>\number\headsep</code>
	footskip	thepage	<code>\number\footskip</code>
	marginparwidth	thepage	<code>\number\marginparwidth</code>
	marginparsep	thepage	<code>\number\marginparsep</code>
	columnwidth	thepage	<code>\number\columnwidth</code>
	columnsep	thepage	<code>\number\columnsep</code>
perpage	pagevalue	perpage	0
	page	perpage	<code><empty></code>
	abspage	perpage	0
savepos	posx	savepos	0
	posy	savepos	0
titleref	title	main	<code><empty></code>
xr	anchor		<code><empty></code>
	externaldocument		<code><empty></code>
	theotype		<code><empty></code>
	title		<code><empty></code>
	url		<code><empty></code>

2.8 Wrapper for advanced situations

```
\zref@wrapper@babel {...} {\<name>}
```

This macro helps to add shorthand support. The second argument is protected, then the code of the first argument is called with the protected name appended. Examples are in the sources.

¹Module `pagelayout` only defines properties if the parameter exists.

`\zref@wrapper@immediate {...}`

There are situations where a label must be written instantly to the `.aux` file, for example after the last page. If the `\zlabel` or `\label` command is put inside this wrapper, immediate writing is enabled. See the implementation for module `lastpage` for an example of its use.

`\zref@wrapper@unexpanded {...}`

Assuming someone wants to extract a value for property `bar` and store the result in a macro `\foo` without traces of the expanding macros and without expanding the value. This (theoretical?) problem can be solved by this wrapper:

```
zref@wrapper@unexpanded{%
  \edef\foo{%
    \zref@extract{someref}{bar}%
  }%
```

The `\edef` forces the expansion of `\zref@extract`, but the extraction of the value is prevented by the wrapper that uses ε -TeX' `\unexpanded` for this purpose. Supported macros are `\zref@extract`, `\zref@extractdefault` and since version 2010/04/22 v2.14 macro `\zref@getcurrent`.

2.9 Counter for unique names

Some modules (`titleref` and `dotfillmin`) need unique names for automatically generated label names.

`\zref@require@unique`

This command creates the unique counter `zref@unique` if the counter does not already exist.

`\thezref@unique`

This command is used to generate unique label names.

3 User interface

3.1 Module `user`

The user interface for this package and its modules is enabled by `zref`'s package option `user` or package `zref-user`. The names of user commands are prefixed by `z` in order to avoid name clashes with existing macros of the same functionality. Thus the package does not disturb the traditional reference scheme, both can be used together.

The syntax descriptions contain the following markers that are intended as hints for programmers:

- | | |
|---------------------|--|
| <code>babel</code> | Babel shorthands are allowed. |
| <code>robust</code> | Robust macro. |
| <code>exp</code> | Expandable version: <ul style="list-style-type: none">• robust, unless the extracted values are fragile,• no babel shorthand support. |
| <code>exp2</code> | Expandable like <code>exp</code> and: <ul style="list-style-type: none">• expandable in exact two steps. |

The basic user interface of the package without modules are commands that mimic the standard L^AT_EX behaviour of `\label`, `\ref`, and `\pageref`:

`\zlabel {⟨refname⟩}{}babel`

Similar to `\label`. It generates a label with name `⟨refname⟩` in the new reference scheme. It is recommended not to use it in moving arguments, so e.g. in the argument of a `\section` or `\caption`. While L^AT_EX removes a normal `\label` when writing to an auxiliary file, it doesn't know about `\zlabel` and one can get multiplied defined labels if the command is executed again, e.g., in the list of figure. It is possible to execute the command together with a standard `\label` by adding it to the label hook with

```
\AddToHookWithArguments{label}{\zlabel{#1}}
```

`\zkvlabel {⟨key value list⟩}{⟨refname⟩}`

Similar to `\zlabel` this creates a label. The `⟨key value list⟩` allows to select the stored properties. The following keys are known:

prop This selects a property that should be used by the label. It can be used more than once. So this will store both properties:

```
\zkvlabel{prop=page,prop=default}{blub}
```

list This selects a list from which then all properties are stored.

delprop This allows to remove a property that has been already selected. So this will not store the page:

```
\zkvlabel{list=main,delprop=page}{blub}
```

immediate This forces properties that are normally delayed and stored only at shipout like `page` to be stored directly.

`\zref [⟨propname⟩] {⟨refname⟩}{}babel`

Without optional argument similar to `\ref`, it returns the default reference property. This property is named **default**:

$$\backslash\mathrm{zref}\{x\} \equiv \backslash\mathrm{zref}[\mathrm{default}]\{x\}$$

`\zpageref {⟨refname⟩}{}babel`

Convenience macro, similar to `\pageref`.

$$\backslash\mathrm{zpageref}\{x\} \equiv \backslash\mathrm{zref}[\mathrm{page}]\{x\}$$

`\zrefused {⟨refname⟩}{}babel`

Some of the user commands in the modules are expandable. The use of such commands do not cause any undefined reference warnings, because inside of expandable contexts this is not possible. However, if there is a place outside of expandable contexts, `\refused` is strongly recommended. The reference `⟨refname⟩` is marked as used, undefined ones will generate warnings.

3.2 Module `abspage`

A new property `abspage` is defined and added to the main property list. It makes use of the LaTeX count `\ReadonlyShipoutCounter` to keep track of the page numbers.

Thus you can reference the absolute page number:

```
Section \zref{foo} is on page \zpageref{foo}.
This is page \zref[abspage]{foo}
of \zref[abspage]{LastPage}.
```

The example also makes use of module `lastpage`.

For compatibility also a counter `abspage` is provided which is increased at shipout too. For technical and historical reasons this counter is zero based: if you use it directly on the first page, e.g with `\arabic{abspage}` you will get 0 as value. (When using `\zref` to retrieve the `abspage` property the first page will be page 1 as expected.). It must be noted that the `perpage` package also provides a `abspage` counter which is *not* zero based but gives 1 on the first page if `zref-abspage` is not loaded or loaded after `perpage`.

3.3 Module `lastpage`

Provides the functionality of package `lastpage` [3] in the new reference scheme. The label `LastPage` is put at the end of the document. You can refer the last page number with:

```
\zref@extract{LastPage}{page} (+ \zref@refused{LastPage})
```

or

```
\zpageref{LastPage} (module user)
```

Since version 2008/10/01 v2.3 the module defines the list `LastPage`. In addition to the properties of the main list label `LastPage` also stores the properties of this list `LastPage`. The default of this list is empty. The list can be used by the user to add additional properties for label `LastPage`.

3.3.1 Tests for last page

Since version 2010/03/26 v2.8 the macros `\zref@iflastpage` and `\ziflastpage` were added. They test the reference, whether it is a reference of the last page.

`\zref@iflastpageexp {<refname>} {<then>} {<else>}`

Macro `\zref@iflastpage` compares the references `<refname>` with `<LastPage>`. Basis of the comparison is the value of property `abspage`, because the values are different for different pages. This is not ensured by property `page`. Therefore module `abspage` is loaded by module `lastpage`. If both values of property `abspage` are present and match, then `<then>` is executed, otherwise code `<else>` is called. If one or both references are undefined or lack the property `abspage`, then `<else>` is executed.

Macro `\zref@iflastpage` is expandable, therefore `\zref@refused` should be called on `<refname>` and `<LastPage>`.

`\ziflastpage {<refname>} {<then>} {<else>}`

Macro `\ziflastpage` has the same function as `\zref@iflastpage`, but adds support for babel shorthands in `<refname>` and calls `\zref@refused`. However macro `\ziflastpage` is not expandable.

3.3.2 Example

```

1 <*example-lastpage>
2 %<END_EXAMPLE
3 \NeedsTeXFormat{LaTeX2e}
4 \documentclass{report}
5
6 \newcounter{foo}
7 \renewcommand*{\thefoo}{\Alph{foo}}
8
9 \usepackage{zref-lastpage,zref-user}[2019/11/29]
10
11 \makeatletter
12 \zref@newprop{thefoo}{\thefoo}
13 \zref@newprop{valuefoo}{\the\value{foo}}
14 \zref@newprop{chapter}{\thechapter}
15 \zref@addprops{LastPage}{thefoo,valuefoo,chapter}
16 \makeatother
17
18 \newcommand*{\foo}{%
19   \stepcounter{foo}%
20   [Current foo: \thefoo]%
21 }
22
23 \begin{document}
24   \chapter{First chapter}
25   Last page is \zref{LastPage}.\
26   Last chapter is \zref[chapter]{LastPage}.\
27   Last foo is \zref[thefoo]{LastPage}.\
28   Last value of foo is \zref[valuefoo]{LastPage}.\
29   \foo
30   \chapter{Second chapter}
31   \foo\foo\foo
32   \chapter{Last chapter}
33   \foo
34 \end{document}
35 %END_EXAMPLE
36 </example-lastpage>

```

3.4 Module thepage

This module `thepage` loads module `abspage`, constructs a reference name using the absolute page number and remembers property `page`. Other properties can be added by adding them to the property list `thepage`.

`\zthepage {<absolute page number>}`

Macro `\zthepage` is basically a `\zpageref`. The reference name is yield by the `<absolute page number>`. If the reference is not defined, then the default for property `page` is used.

`\zref@thepage@nameexp {<absolute page number>}`

Macro `\zref@thepage@name` returns the internal reference name that is constructed using the `<absolute page number>`. The internal reference name should not be used directly, because it might change in future versions.

<code>\zref@thepage^{exp} {\absolute page number}</code> <code>\zref@thepage@refused {\absolute page number}</code>
--

Macro `\zref@thepage` returns the page number (`\thepage`) of `\absolute page number`. Because this macro is expandable, `\zref@thepage@refused` is used outside an expandable context to mark the reference as used.

3.5 Module `nextpage`

<code>\znextpage</code>

Macro `\znextpage` prints `\thepage` of the following page. It gets the current absolute page number by using a label. There are three cases for the next page:

1. The next page is not known yet because of undefined references. Then `\unknownnextpagename` is used instead. The default for this macro is the default of property `page`.
2. This page is the last page. Then `\znonextpagename` is used. Its default is empty.
3. The next page is known, then `\thepage` of the next page is used (the value of property `page` of the next page).

3.5.1 Configuration

The behaviour can be configured by the following macros.

<code>\unknownnextpagename</code> <code>\znonextpagename</code>
--

If the next page is not known or available, then `\znextpage` uses these name macros as default. `\unknownnextpagename` is used in case of undefined references. Default is the value of property `page` of the next page (`\thepage`). Module `thepage` is used.

Macro `\znonextpagename` is used, if the next page does not exists. That means that the current page is last page. The default is empty.

<code>\znextpagesetup {\unknown} {\no next} {\next}</code>
--

According to the case (see `\znextpage`) macro `\znextpage` calls an internal macro with an argument. The argument is either `\thepage` of the next page or one of `\unknownnextpagename` or `\znonextpagename`. These internal macro can be changed by `\znextpagesetup`. It expects the definition texts for these three cases of a macro with one argument. The default is

```
\znextpagesetup{#1}{#1}{#1}
```

3.5.2 Example

```

37 \example-nextpage)
38 %<<END_EXAMPLE
39 \documentclass{book}
40
41 \usepackage{zref-nextpage}[2019/11/29]
42 \znextpagesetup
43   {\thepage}% next page is unknown
44   {\thepage\ (#1)}% this page is last page
45   {\thepage\ $\rightarrow$ #1}% next page is known
46 \renewcommand*{\znonextpagename}{last page}
47

```

```

48 \usepackage{fancyhdr}
49 \pagestyle{fancy}
50 \fancyhf{}
51 \fancyhead[LE,RO]{\znexthpage}
52 \fancypagestyle{plain}{%
53   \fancyhf{}%
54   \fancyhead[LE,RO]{\znexthpage}%
55 }
56
57 \begin{document}
58 \frontmatter
59   \tableofcontents
60 \mainmatter
61   \chapter{Hello World}
62   \clearpage
63   \section{Last section}
64 \end{document}
65 %END_EXAMPLE
66 \example-nextpage

```

3.6 Module **totpages**

For the total number of pages of a document you need to know the absolute page number of the last page. Both modules **abspage** and **lastpage** are necessary and automatically enabled.

`\ztotpagesexp`

Prints the total number of pages or 0 if this number is not yet known. It expands to an explicit number and can also be used even in expandable calculations (`\numexpr`) or counter assignments.

3.7 Module **pagelayout**

The module defines additional properties for each parameter of the page layout that is effective during page shipout. The value of length parameters is given in sp without the unit as plain number.

Some parameters are specific for a class (e.g. **stockwidth** and **stockheight** for class **memoir**) or the $\text{\TeX}$ engine like pdf $\text{\TeX}$. If the parameter is not available, then the property will not be defined. The default value of the property is the current setting of the parameter.

The module **thepage** is loaded that generates a label for each page. The properties of module **pagelayout** are added to the property list **thepage** of module **thepage**.

List of properties:

parameter	property	remarks
<code>\mag</code>	mag	
<code>\paperwidth</code>	paperwidth	
<code>\paperheight</code>	paperheight	
<code>\stockwidth</code>	stockwidth	class memoir
<code>\stockheight</code>	stockheight	class memoir
<code>\pdfpagewidth</code>	pdfpagewidth	pdfTeX, LuaTeX
<code>\pdfpageheight</code>	pdfpageheight	pdfTeX, LuaTeX
<code>\pdfhorigin</code>	pdfhorigin	pdfTeX, LuaTeX
<code>\pdfvorigin</code>	pdfvorigin	pdfTeX, LuaTeX
<code>\hoffset</code>	hoffset	
<code>\voffset</code>	voffset	
<code>\topmargin</code>	topmargin	
<code>\oddsidemargin</code>	oddsidemargin	
<code>\evensidemargin</code>	evensidemargin	
<code>\textwidth</code>	textwidth	
<code>\textheight</code>	textheight	
<code>\headheight</code>	headheight	
<code>\headsep</code>	headsep	
<code>\footskip</code>	footskip	
<code>\marginparwidth</code>	marginparwidth	
<code>\marginparsep</code>	marginparsep	
<code>\columnwidth</code>	columnwidth	
<code>\columnsep</code>	columnsep	

`\zlistpagelayout`

At the end of document the page layout parameter for each page are printed into the `.log` file if macro `\zlistpagelayout` is called before `\end{document}` (preamble is a good place).

3.8 Module marks

ToDo.

3.9 Module runs

Module `runs` counts the $\text{\LaTeX}$ runs since last `.aux` file creation and prints the number in the `.log` file.

`\zrunsexp`

Prints the the total number of $\text{\LaTeX}$ runs including the current one. It expands to an explicit number. Before `\begin{document}` the value is zero meaning the `.aux` file is not read yet. If a previous `.aux` file exists, the value found there increased by one is the new number. Otherwise `\zruns` is set to one. $\text{\LaTeX}$ runs where the `.aux` files are not rewritten are not counted (see `\nofiles`).

3.10 Module perpage

With `\@addtoreset` or `\numberwithin` a counter can be reset if another counter is incremented. This do not work well if the other counter is the page counter. The page counter is incremented in the output routine that is often called asynchronous somewhere on the next page. A reference mechanism costs at least two $\text{\LaTeX}$ runs, but ensures correct page counter values.

`\zmakeperpage [reset] {counter}`

At the of a new page counter *counter* starts counting with value *reset* (default is 1). The macro has the same syntax and semantics as `\MakePerPage` of package `perpage` [5]. Also `perpage` of package `footmisc` [1] can easily be simulated by

```
\zmakeperpage{footnote} % \usepackage[perpage]{footmisc}
```

If footnote symbols are used, some people dislike the first symbol †. It can easily be skipped:

```
\zmakeperpage[2]{footnote}
```

`\thezpage`
`counter zpage`

If the formatted counter value of the counter that is reset at a new page contains the page value, then you can use `\thezpage`, the page number of the current page. Or counter `zpage` can be used, if the page number should be formatted differently from the current page number. Example:

```
newcounter{foobar}  
zmakeperpage{foobar}  
renewcommand*{\thefoobar}{\thezpage-\arabic{foobar}}  
% or  
renewcommand*{\thefoobar}{\roman{zpage}-\arabic{foobar}}
```

`\zunmakeperpage {counter}`

The reset mechanism for this counter is deactivated.

3.11 Module **counter**

This option just add the property `counter` to the main property list. The property stores the counter name, that was responsible for the reference. This is the property `hyperref`'s `\autoref` feature uses. Thus this property `counter` may be useful for a reimplementaion of the `autoref` feature, see the section 4 with the `todo` list.

3.12 Module **titleref**

This option makes section and caption titles available to the reference system similar to packages `titleref` or `nameref`.

The module patches a number of commands to store the title in an internal variable which is then exported by `\zlabel`.

If `\DocumentMetadata` is used, most of the patches are no longer functional, the internal variable is no longer used. Instead `\zlabel` will store the current value of `\@currentlabelname`. This works with heading commands, captions, and with L^AT_EX 2026-11-01 will also work with theorems and item of description lists.

`\ztitleref {refname}`^{babel}

Print the section or caption title of reference *refname*, similar to `\nameref` or `\titleref`.

`\ztitlerefsetup {key1=value1, key2=value2, ...}`

This command allows to configure the behaviour of module `titleref`. The following keys are available:

`title=<value>`

Sets the current title.

`stripperperiod=true|false`

Follow package `nameref` that removes a last period. Default: `true`.

`expand=true|false`

Package `\titleref` expands the title first. This way garbage and dangerous commands can be removed, e.g. `\label`, `\index...`. See implementation section for more details. Default is `false`.

`cleanup={...}`

Hook to add own cleanup code, if method `expand` is used. See implementation section for more details.

3.13 Module `savepos`

This option supports a feature that all engines nowadays provide: (and $\text{\TeX}$). They are able to tell the current position on the page. The page position is not instantly known. First the page must be constructed by $\text{\TeX}$'s asynchronous output routine. Thus the time where the position is known is the page shipout time. Thus a reference system where the information is recorded in the first run and made available for use in the second run comes in handy.

`\zsavapos {<refname>}`

It generates a reference with name `<refname>`. The reference stores the location where `\zsavapos` is executed in properties `posx` and `posy`.

`\zsavaposx {<refname>}`
`\zsavaposy {<refname>}`

Same as `\zsavapos` except that only the x or y component of the position is stored. Since 2011/12/05 v2.23.

`\zposxexp {<refname>}`
`\zposyexp {<refname>}`

Get the position as number. Unit is sp. Horizontal positions by `\zposx` increase from left to right. Vertical positions by `\zposy` from bottom to top.

Do not rely on absolute page numbers. Because of problems with the origin the numbers may differ in DVI or PDF mode of $\text{\pdfTeX}$. Therefore work with relative values by comparisons.

Both `\zposx` and `\zposy` are expandable and can be used inside calculations (`\setcounter`, `\addtocounter`, package `calc`, `\numexpr`). However this property prevents from notifying $\text{\LaTeX}$ that the reference is actually used (the notifying is not expandable). Therefore you should mark the reference as used by `\zrefused`.

This module uses $\text{\pdfTeX}$'s `\pdfsavepos`, `\pdflastxpos`, and `\pdflastypos` and similar commands in the other engines. They are available in PDF mode and since $\text{\pdfTeX}$ version 1.40.0 also in DVI mode.

`\zref@savepos`

Macro `\zref@savepos` performs the first part of `\zsavapos` by calling `\tex_savepos:D` (if `.aux` files are writable).

Thus `\zsavapos` is basically `\zref@savepos` followed by `\zref@labelbylist{<refname>}{savepos}`. `\savepos` also adds `\zref@savepos` at the end to support `\beginR` where the whatits

are processed in reverse order. The property list `savepos` contains the properties `posx` and `posy`.

3.14 Module **abspos**

Module **abspos** allows to get various values of the page layout. There is no user command, only a number of internal commands. For example:

```
\zref@absposx{<label>}{<value>}{<position>}
\zref@absposy{<label>}{<value>}{<position>}
```

The return value is like in the module **savepos** a number representing a length in sp. The length are measured from the bottom left of the page.

`<label>` is a label set with `\zlabel` or `\zsavepos` that allows to retrieve the absolute page number.

`<position>` is for the x-command one of **left**, **right** or **center**. For the y-command it is one of **top**, **bottom**, **center**.

The possible content of `<value>` can be seen in the following table. Be aware that the code makes some assumptions which are perhaps not always true – for example that the left of the head is identical to the left of the body.

value	axis	comments
media	x	left=0, right= <code>\pdfpagewidth</code>
paper	x	left=0, right= <code>\paperwidth</code>
stock	x	derived from paper
media	y	bottom=0, top= <code>\pdfpageheight</code>
paper	y	top= <code>\pdfpageheight</code> , bottom=top- <code>\paperheight</code>
stock	y	top derived from paper
head	x	calculated with <code>hoffset</code> , <code>horigin</code> , etc
head	y	calculated
body	x	= head value
body	y	= head bottom - <code>\headsep</code>
foot	x	= head
foot	y	calculated from body bottom and <code>\footskip</code>
marginpar	x	different on odd/even pages!
marginpar	y	= body

3.15 Module **dotfill**

`\zdotfill`

This package provides the command `\zdotfill` that works similar to `\dotfill`, but can be configured. Especially it suppresses the dots if a minimum number of dots cannot be set.

`\zdotfillsetup {key1=value1, key2=value2, ...}`

This command allows to configure the behaviour of `\zdotfill`. The following keys are available:

`min=<count value>`

If the actual number of dots are smaller than `<count value>`, then the dots are suppressed. Default: 2.

`unit=<dimen value>`

The width of a dot unit is given by `<dimen value>`. Default: 0.44em (same as the unit in `\dotfill`).

`dot=<value>`

The dot itself is given by `<value>`. Default: . (dot, same as the dot in `\dotfill`).

3.16 Module `env`

This module defines two properties `envname` and `envline`. They remember the name of the environment and the line number (in the source) at the start of the environment.

3.17 Module `xr`

This package provides the functionality of package `xr`, see [8]. It also supports the syntax of `xr-hyper`.

`\xexternaldocument * [⟨prefix⟩]babel {⟨external document⟩} [⟨url⟩]`

See `\xexternaldocument` for a description of this option. The found labels also get a property `externaldocument` that remembers `⟨external document⟩`. The standard reference scheme and the scheme of this package use different name spaces for reference names. If the external document uses both systems. Then one import statement would put the names in one namespace and probably causing problems with multiple references of the same name. Thus the star form only looks for `\newlabel` in the `.aux` files, whereas without star only `\zref@newlabels` are used.

In the star form it tries to detect labels from `hyperref`, `titleref`, and `ntheorem`. If such an extended property from the packages before cannot be found or are empty, they are not included in the imported reference.

Warnings are given if a reference name is already in use and the item is ignored. Unknown properties will automatically be declared.

If the external references contain `anchor` properties, then we need also a url to be able to address the external file. As default the filename is taken with a default extension.

`\zxrsetup {key1=value1, key2=value2, ...}`

The following setup options are available:

ext: It sets the default extension.

tozreflabel: Boolean option. The found references are imported as zref labels. This is enabled by default.

toltxlabel: Boolean option. The found references are imported as L^AT_EX labels. Packages `nameref`, `titleref` and class `memoir` are supported.

urluse: Boolean option. If enabled, then a URL is stored in a macro and the macro is put in property ‘urluse’. The URL is not put in property ‘url’. The purpose is to save T_EX memory.

verbose: Boolean option. List the imported labels in the `.log` file. Default is `false`.

`\zref@xr@ext`

If the `⟨url⟩` is not specified in `\zref@externaldocument`, then the url will be constructed with the file name and this macro as extension. `\XR@ext` is used if `hyperref` is loaded, otherwise `pdf`.

3.18 Module `pageattr`

This module allows to recover the content of the register `\pdfpageattr` and `\pdfpagesattr` in `pdftex` and the equivalent register in `luatex`. There is no user command. Programmer commands are

```

\zref@pdfpageattr{\absolute page number}}
\zref@pdfpagesattr{\absolute page number}}
\zref@pdfpagesattr@used

```

Note that if the PDF management is used, page attributes are often set only at shipout.

4 ToDo

Among other things the following issues are left for future work:

- Other applications: autoref, hyperref, ...

5 Example

```

67 \example
68 \documentclass{book}
69
70 \usepackage[ngerman]{babel}%
71
72 \usepackage[savepos,totpages,titleref,dotfill,counter,user]{zref}
73

```

Chapters are wrapped inside `\ChapterStart` and `\ChapterStop`. The first argument `#1` of `\ChapterStart` is used to form a label id `chap:#1`. At the end of the chapter another label is set by `\zref@wrapper@immediate` in older LaTeX formats, because otherwise at the end of document a deferred write would not be written, because there is no page for shipout.

Also this example shows how chapter titles can be recorded. A new property `chaptitle` is declared and added to the main property list. In `\ChapterStart` the current value of the property is updated.

```

74 \makeatletter
75 \zref@newprop{chaptitle}{}
76 \zref@addprop{main}{chaptitle}
77
78 \newcommand*{\ChapterStart}[2]{%
79   \cleardoublepage
80   \def\current@chapid{#1}%
81   \zref@setcurrent{chaptitle}{#2}%
82   \chapter{#2}%
83   \zlabel{chap:#1}%
84 }
85 \newcommand*{\ChapterStop}{%
86   \cleardoublepage
87   \zref@wrapper@immediate{%
88     \zref@labelbyprops{chapend:\current@chapid}{abspage}%
89   }%
90 }

```

`\ChapterPages` calculates and returns the number of pages of the referenced chapter.

```

91 \newcommand*{\ChapterPages}[1]{%
92   \zrefused{chap:#1}%
93   \zrefused{chapend:#1}%
94   \number\numexpr
95     \zref@extract{chapend:#1}{abspage}%
96     -\zref@extract{chap:#1}{abspage}%
97     +1\relax
98 }
99 \makeatother

```

```
100 \begin{document}
```

As exception we use `\makeatletter` here, because this is just an example file that also should show some of programmer's interface.

```
101 \makeatletter
102
103 \frontmatter
104 \zlabel{documentstart}
105
106 \begin{itemize}
107 \item
108   The frontmatter part has
109   \number\numexpr\zref@extract{chap:first}{abspage}-1\relax
110   ~pages.
111 \item
112   Chapter \zref{chap:first} has \ChapterPages{first} page(s).
113 \item
114   Section \zref{hello} is on the
115   \ifcase\numexpr
116     \zref@extractdefault{hello}{page}{0}%
117     -\zref@extractdefault{chap:first}{page}{0}%
118     +1\relax
119     ??\or first\or second\or third\or forth\fi
120   ~page inside its chapter.
121 \item
122   The document has
123   \zref[abspage]{LastPage} pages.
124   This number is \ifodd\ztotpages odd\else even\fi.
125 \item
126   The last page is labeled with \zpageref{LastPage}.
127 \item
128   The title of chapter \zref{chap:next} %
129   is ``\zref[chaptitle]{chap:next}''.
130 \end{itemize}
131
132 \tableofcontents
133
134 \mainmatter
135 \ChapterStart{first}{First chapter}
136
```

The user level commands should protect babel shorthands where possible. On the other side, expandable extracting macros are useful in calculations, see above the examples with `\numexpr`.

```
137 \section{Test}
138 \zlabel{a"o}
139 Section \zref{a"o} on page
140 \zref@wrapper@babel\zref@extract{a"o}{page}.
141
142 Text.
143 \newpage
144
145 \section{Hello World}
146 \zlabel{hello}
147
148 \ChapterStop
149
150 \ChapterStart{next}{Next chapter with \emph{umlauts}: "a"o"u"s}
151
```

Here an example follows that makes use of the “savepos” feature. The position on the page is not known before the page is constructed and shipped out. Therefore the position

ist stored in references and are available for calculations in the next $\text{\LaTeX}$ compile run.

```

152 The width of the first column is
153 \the\dimexpr \zposx{secondcol}sp - \zposx{firstcol}sp\relax,\,
154 the height difference of the two baselines is
155 \the\dimexpr \zposy{firstcol}sp - \zposy{secondline}sp\relax:\,
156 \begin{tabular}{ll}
157 \zsavepos{firstcol}Hello&\zsavepos{secondcol}World\\
158 \zsavepos{secondline}Second line&foobar\\
159 \end{tabular}
160

```

With $\text{\zrefused}$ $\text{\LaTeX}$ is notified, if the references are not yet available and $\text{\LaTeX}$ can generate the rerun hint.

```

161 \zrefused{firstcol}
162 \zrefused{secondcol}
163 \zrefused{secondline}
164
165 \ChapterStop

```

Test for module $\text{\dotfill}$.

```

166 \ChapterStart{dotfill}{Test for dotfill feature}
167 \newcommand*{\dfptest}[1]{%
168   #1&
169   [\makebox[{#1}]{\dotfill}]&
170   [\makebox[{#1}]{\zdotfill}]\,
171 }
172 \begin{tabular}{rll}
173 & [\verb|\dotfill|] & [\verb|\zdotfill|]\,
174 \dfptest{0.43em}
175 \dfptest{0.44em}
176 \dfptest{0.45em}
177 \dfptest{0.87em}
178 \dfptest{0.88em}
179 \dfptest{0.89em}
180 \dfptest{1.31em}
181 \dfptest{1.32em}
182 \dfptest{1.33em}
183 \end{tabular}
184 \ChapterStop
185 \end{document}
186

```

6 Implementation

6.1 Package $\text{\zref}$

6.1.1 Identification

```

187 <*package>
188 \NeedsTeXFormat{LaTeX2e}
189 \ProvidesPackage{zref}
190 [2026-09-16 v2.38 A new reference scheme for LaTeX (HO)]%

```

6.1.2 Load basic module

```

191 \RequirePackage{zref-base}[2019/11/29]

```

Abort package loading if $\text{\zref-base}$ could not be loaded successfully.

```

192 \@ifundefined{ZREF@base@ok}{\endinput}{}

```

6.1.3 Process options

Known modules are loaded and the release date is checked.

```
193 \def\ZREF@temp#1{%
194   \DeclareOption{#1}{%
195     \AtEndOfPackage{%
196       \RequirePackage{zref-#1}[2019/11/29]%
197     }%
198   }%
199 }
200 \ZREF@temp{abspage}
201 \ZREF@temp{counter}
202 \ZREF@temp{dotfill}
203 \ZREF@temp{hyperref}
204 \ZREF@temp{lastpage}
205 \ZREF@temp{marks}
206 \ZREF@temp{nextpage}
207 \ZREF@temp{pageattr}
208 \ZREF@temp{pagelayout}
209 \ZREF@temp{perpage}
210 \ZREF@temp{runs}
211 \ZREF@temp{savepos}
212 \ZREF@temp{thepage}
213 \ZREF@temp{titleref}
214 \ZREF@temp{totpages}
215 \ZREF@temp{user}
216 \ZREF@temp{xr}

217 \ProcessOptions\relax
218 \</package>
```

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:

\zref@: Macros of the programmer's interface.

\ZREF@: Internal macros.

\Z@L@listname: The properties of the list *<listname>*.

\Z@D@propname: The default value for property *<propname>*.

\Z@E@propname: Extract function for property *<propname>*.

\Z@X@propname: Information whether a property value for property *<propname>* is expanded immediately or at shipout time.

\Z@C@propname: Current value of the property *<propname>*.

\Z@R@labelname: Data for reference *<labelname>*.

\ZREF@org@: Original versions of patched commands.

\z: For macros in user land, defined if module *user* is set.

The following family names are used for keys defined according to the *keyval* package:

ZREF@TR: Setup for module *titleref*.

6.2.2 Identification

```

219 <*base>
220 \NeedsTeXFormat{LaTeX2e}
221 \ProvidesPackage{zref-base}%
222 [2026-09-16 v2.38 Module base for zref (H0)]%
```

6.2.3 Utilities

```

223 \providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
224 \RequirePackage{ltxcmds}[2010/12/02]
225 \RequirePackage{infwarerr}[2010/04/08]
226 \RequirePackage{kvsetkeys}[2010/03/01]
227 \RequirePackage{kvdefinekeys}[2010/03/01]
228 \RequirePackage{pdftexcmds}[2010/04/01]
```

`\ZREF@name` Several times the package name is used, thus we store it in `\ZREF@name`.

```

229 \def\ZREF@name{zref}
```

(End of definition for \ZREF@name.)

```

230 \ltx@ifundefined{protected}{%
231 \RequirePackage{makerobust}[2006/03/18]%
```

`\ZREF@Robust`

```

232 \def\ZREF@Robust#1#2{%
233 \def\ZREF@temp{\MakeRobustcommand#2}%
234 \afterassignment\ZREF@temp
235 #1#2%
236 }%
```

(End of definition for \ZREF@Robust.)

```

237 }{%
```

`\ZREF@Robust`

```

238 \def\ZREF@Robust#1{%
239 \protected#1%
240 }%
```

(End of definition for \ZREF@Robust.)

```

241 }
```

`\ZREF@IfDefinable`

```

242 \def\ZREF@IfDefinable#1#2#3{%
243 \@ifdefinable{#1}{%
244 \ZREF@Robust{#2}#1#3%
245 }%
246 }
```

(End of definition for \ZREF@IfDefinable.)

`\ZREF@UpdatePdfTeX` `\ZREF@UpdatePdfTeX` is used as help message text in error messages.

```

247 \def\ZREF@UpdatePdfTeX{Update pdfTeX.}
```

(End of definition for \ZREF@UpdatePdfTeX.)

`\ifZREF@found` The following switch is used in list processing.

```

248 \newif\ifZREF@found
```

(End of definition for \ifZREF@found.)

`\ZREF@patch` Macro `\ZREF@patch` first checks the existence of the command and safes it.

```

249 \def\ZREF@patch#1{%
250   \ltx@ifundefined{#1}{%
251     \ltx@gobble
252   }{%
253     \expandafter\let\csname ZREF@org@#1\expandafter\endcsname
254     \csname #1\endcsname
255     \ltx@firstofone
256   }%
257 }

```

(End of definition for `\ZREF@patch`.)

6.2.4 Check for ε -TeX

The use of ε -TeX should be standard nowadays for L^AT_EX. We test for ε -TeX in order to use its features later.

```

258 \ltx@ifundefined{eTeXversion}{%
259   \PackageError\ZREF@name{%
260     Missing support for eTeX; package is abandoned%
261   }{%
262     Use a TeX compiler that support eTeX and enable eTeX %
263     in the format.%
264   }%
265   \endinput
266 }{}%

267 \RequirePackage{etexcmds}[2007/09/09]
268 \ifetex@unexpanded
269 \else
270   \PackageError\ZREF@name{%
271     Missing e-TeX's \string\unexpanded.\MessageBreak
272     Add \string\RequirePackage\string{etexcmds\string} before %
273     \string\documentclass%
274   }{%
275     Probably you are using some package (e.g. ConTeXt) that %
276     redefines \string\unexpanded%
277   }%
278   \expandafter\endinput
279 \fi

```

6.2.5 Auxiliary file stuff

We are using some commands in the `.aux` files. However sometimes these auxiliary files are interpreted by L^AT_EX processes that haven't loaded this package (e.g. package `xr`). Therefore we provide dummy definitions.

```

280 \RequirePackage{auxhook}
281 \AddLineBeginAux{%
282   \string\providecommand\string\zref@newlabel[2]{}%
283 }

```

`\ZREF@RefPrefix`

```

284 \def\ZREF@RefPrefix{Z@R}

```

(End of definition for `\ZREF@RefPrefix`.)

`\zref@newlabel` For the implementation of `\zref@newlabel` we call the same internal macro `\@newl@bel` that is used in `\newlabel`. Thus we have for free:

- `\Z@R@labelname` is defined.
- L^AT_EX's check for multiple references.
- L^AT_EX's check for changed references.

```

285 \ZREF@Robust\edef\zref@newlabel{%
286   \noexpand\@newl@bel{\ZREF@RefPrefix}%
287 }

```

(End of definition for \zref@newlabel. This function is documented on page 8.)

6.2.6 Property lists

\zref@newlist Property lists are stored as list of property names enclosed in curly braces. **\zref@newlist** creates a new list as empty list. Assignments to property lists are global.

```

288 \ZREF@Robust\def\zref@newlist#1{%
289   \zref@iflistundefined{#1}{%
290     \ifdefinable{Z@L@#1}{%
291       \global\expandafter\let\csname Z@L@#1\endcsname\ltx@empty
292       \PackageInfo\ZREF@name{New property list: #1}%
293     }%
294   }{%
295     \PackageError\ZREF@name{%
296       Property list `#1' already exists%
297     }\@ehc
298   }%
299 }

```

(End of definition for \zref@newlist. This function is documented on page 6.)

\zref@iflistundefined **\zref@iflistundefined** checks the existence of the property list #1. If the property list is present, then #2 is executed and #3 otherwise.

```

300 \def\zref@iflistundefined#1{%
301   \ltx@ifundefined{Z@L@#1}%
302 }

```

(End of definition for \zref@iflistundefined. This function is documented on page 6.)

\zref@listexists **\zref@listexists** only executes #2 if the property list #1 exists and raises an error message otherwise.

```

303 \ZREF@Robust\def\zref@listexists#1{%
304   \zref@iflistundefined{#1}{%
305     \PackageError\ZREF@name{%
306       Property list `#1' does not exist%
307     }\@ehc
308   }%
309 }

```

(End of definition for \zref@listexists. This function is documented on page 6.)

\zref@iflistcontainsprop **\zref@iflistcontainsprop** checks, whether a property #2 is already present in a property list #1.

```

310 \ZREF@Robust\def\zref@iflistcontainsprop#1#2{%
311   \zref@iflistundefined{#1}{%
312     \ltx@secondoftwo
313   }{%
314     \begingroup\expandafter\endgroup
315     \expandafter\in@
316     \csname#2\expandafter\expandafter\expandafter\endcsname
317     \expandafter\expandafter\expandafter{\csname Z@L@#1\endcsname}%
318     \csname \ltx@ifin@ first\else second\fi oftwo\endcsname
319   }%
320 }

```

(End of definition for \zref@iflistcontainsprop. This function is documented on page 7.)

`\zref@listforloop`

```

321 \def\zref@listforloop#1#2{%
322   \zref@listexists{#1}{%
323     \expandafter\expandafter\expandafter\@tfor
324     \expandafter\expandafter\expandafter\zref@prop
325     \expandafter\expandafter\expandafter:%
326     \expandafter\expandafter\expandafter=%
327     \csname Z@L@#1\endcsname
328     \do{%
329       \begingroup
330       \escapechar=-1 %
331       \edef\x{\endgroup
332         \def\noexpand\zref@prop{%
333           \expandafter\string\zref@prop
334         }%
335       }%
336       \x
337       #2\zref@prop
338     }%
339   }%
340 }

```

(End of definition for \zref@listforloop.)

\zref@addprops `\zref@addprop` adds the properties #2 to the property list #1, if the property is not already in the list. Otherwise a warning is given.

```

341 \ZREF@Robust\def\zref@addprops#1#2{%
342   \zref@listexists{#1}{%
343     \comma@parse{#2}{%
344       \zref@propexists\comma@entry{%
345         \zref@iflistcontainsprop{#1}\comma@entry{%
346           \PackageWarning\ZREF@name{%
347             Property '\comma@entry' is already in list `#1'%
348           }%
349         }{%
350           \begingroup\expandafter\endgroup
351           \expandafter\g@addto@macro
352           \csname Z@L@#1\expandafter\endcsname
353           \expandafter{\csname\comma@entry\endcsname}%
354         }%
355       }%
356       \ltx@gobble
357     }%
358   }%
359 }

```

(End of definition for \zref@addprops. This function is documented on page 6.)

\zref@addprop `\zref@addprop` adds the property #2 to the property list #1, if the property is not already in the list. Otherwise a warning is given.

```

360 \ZREF@Robust\def\zref@addprop#1#2{%
361   \zref@listexists{#1}{%
362     \zref@propexists{#2}{%
363       \zref@iflistcontainsprop{#1}{#2}{%
364         \PackageWarning\ZREF@name{%
365           Property `#2' is already in list `#1'%
366         }%
367       }{%
368         \begingroup\expandafter\endgroup
369         \expandafter\g@addto@macro
370         \csname Z@L@#1\expandafter\endcsname
371         \expandafter{\csname#2\endcsname}%
372       }%

```

```

373     }%
374 }%
375 }

```

(End of definition for \zref@addprop. This function is documented on page 6.)

\zref@localaddprops

```

376 \ZREF@Robust\def\zref@localaddprops#1#2{%
377   \zref@listexists{#1}{%
378     \comma@parse{#2}{%
379       \zref@propexists\comma@entry{%
380         \zref@iflistcontainsprop{#1}\comma@entry{%
381           \PackageWarning\ZREF@name{%
382             Property '\comma@entry' is already in list `#1'%
383           }%
384         }{%
385           \begingroup\expandafter\endgroup
386           \expandafter\ltx@LocalAppendToMacro
387           \csname Z@L@#1\expandafter\endcsname
388           \expandafter{\csname\comma@entry\endcsname}%
389         }%
390       }%
391     \ltx@gobble
392   }%
393 }%
394 }

```

(End of definition for \zref@localaddprops. This function is documented on page 6.)

\zref@localaddprop

```

395 \ZREF@Robust\def\zref@localaddprop#1#2{%
396   \zref@listexists{#1}{%
397     \zref@propexists{#2}{%
398       \zref@iflistcontainsprop{#1}{#2}{%
399         \PackageWarning\ZREF@name{%
400           Property `#2' is already in list `#1'%
401         }%
402       }{%
403         \begingroup\expandafter\endgroup
404         \expandafter\ltx@LocalAppendToMacro
405         \csname Z@L@#1\expandafter\endcsname
406         \expandafter{\csname#2\endcsname}%
407       }%
408     }%
409   }%
410 }

```

(End of definition for \zref@localaddprop.)

```

411 \ltx@ifundefined{pdf@strcmp}{%

```

\zref@delprop

```

412 \ZREF@Robust\def\zref@delprop{%
413   \ZREF@delprop\gdef
414 }%

```

(End of definition for \zref@delprop. This function is documented on page 6.)

\zref@localdelprop

```

415 \ZREF@Robust\def\zref@localdelprop{%
416   \ZREF@delprop\def
417 }%

```

(End of definition for \zref@localdelprop. This function is documented on page 6.)

`\ZREF@delprop`

```

418 \def\ZREF@delprop#1#2#3{%
419   \zref@listexists{#2}{%
420     \begingroup
421     \escapechar=-1 %
422     \def\ZREF@param{#3}%
423     \@onelevel@sanitize\ZREF@param
424     \toks@{%
425       \expandafter\expandafter\expandafter\ZREF@@delprop
426       \csname Z@L@#2\endcsname!%
427     \expandafter\endgroup
428     \expandafter#1\csname Z@L@#2\expandafter\endcsname
429     \expandafter{%
430       \the\toks@
431     }%
432   }%
433 }%
```

(End of definition for \ZREF@delprop.)

`\ZREF@@delprop`

```

434 \def\ZREF@@delprop#1{%
435   \expandafter\ZREF@@@delprop\expandafter{\string#1}#1%
436 }%
```

(End of definition for \ZREF@@delprop.)

`\ZREF@@@delprop`

```

437 \def\ZREF@@@delprop#1#2{%
438   \ifx#2!%
439   \else
440     \def\ZREF@temp{#1}%
441     \@onelevel@sanitize\ZREF@temp
442     \ifx\ZREF@param\ZREF@temp
443     \else
444       \toks@\expandafter{%
445         \the\expandafter\toks@\csname#1\endcsname
446       }%
447     \fi
448     \expandafter\ZREF@@delprop
449   \fi
450 }%
```

(End of definition for \ZREF@@@delprop.)

451 }{%

`\zref@delprop`

```

452 \ZREF@Robust\def\zref@delprop{%
453   \ZREF@delprop\xdef
454 }%
```

(End of definition for \zref@delprop. This function is documented on page 6.)

`\zref@localdelprop`

```

455 \ZREF@Robust\def\zref@localdelprop{%
456   \ZREF@delprop\edef
457 }%
```

(End of definition for \zref@localdelprop. This function is documented on page 6.)

`\ZREF@delprop`

```

458 \def\ZREF@delprop#1#2#3{%
459   \zref@iflistexists{#2}{%
460     \def\ZREF@param{#3}%
461     \edef\ZREF@SavedEscapechar{\the\escapechar}%
462     \escapechar=-1 %
463     \expandafter#1\csname Z@L@#2%
464     \expandafter\expandafter\expandafter\endcsname{%
465       \expandafter\expandafter\expandafter\ZREF@@delprop
466       \csname Z@L@#2\endcsname!%
467     }%
468     \escapechar=\ZREF@SavedEscapechar\relax
469   }%
470 }%

```

(End of definition for \ZREF@delprop.)

`\ZREF@@delprop` Caution: #1 might be an `\if` or similar token.

```

471 \def\ZREF@@delprop#1{%
472   \expandafter\ZREF@@@delprop\expandafter{\string#1}#1%
473 }%

```

(End of definition for \ZREF@@delprop.)

`\ZREF@@@delprop`

```

474 \def\ZREF@@@delprop#1#2{%
475   \ifx#2!%
476   \else
477     \ifnum\pdf@strcmp{#1}{\ZREF@param}=\ltx@zero
478     \else
479       \expandafter\noexpand\csname#1\endcsname
480     \fi
481     \expandafter\ZREF@@delprop
482   \fi
483 }%

```

(End of definition for \ZREF@@@delprop.)

484 }

6.2.7 Properties

`\zref@ifpropundefined` `\zref@ifpropundefined` checks the existence of the property #1. If the property is present, then #2 is executed and #3 otherwise.

```

485 \def\zref@ifpropundefined#1{%
486   \ltx@ifundefined{Z@E@#1}%
487 }

```

(End of definition for \zref@ifpropundefined. This function is documented on page 7.)

`\zref@propexists` Some macros rely on the existence of a property. `\zref@propexists` only executes #2 if the property #1 exists and raises an error message otherwise.

```

488 \ZREF@Robust\def\zref@propexists#1{%
489   \zref@ifpropundefined{#1}{%
490     \PackageError\ZREF@name{%
491       Property `#1' does not exist%
492     }\@ehc
493   }%
494 }

```

(End of definition for \zref@propexists. This function is documented on page 7.)

\zref@newprop A new property is declared by `\zref@newprop`, the property name $\langle propname \rangle$ is given in #1. The property is created and configured. If the star form is given, then the expansion of the property value is delayed to page shipout time, when the reference is written to the `.aux` file.

\Z@D@propname: Stores the default value for this property.

\Z@E@propname: Extract function.

\Z@X@propname: Information whether the expansion of the property value is delayed to shipout time.

\Z@C@propname: Current value of the property.

```

495 \ZREF@Robust\def\zref@newprop{%
496   \@ifstar{%
497     \let\ZREF@X\noexpand
498     \ZREF@newprop
499   }{%
500     \let\ZREF@X\ltx@empty
501     \ZREF@newprop
502   }%
503 }
```

(End of definition for \zref@newprop. This function is documented on page 7.)

\ZREF@newprop

```

504 \def\ZREF@newprop#1{%
505   \edef\ZREF@P{#1}%
506   \@onelevel@sanitize\ZREF@P
507   \begingroup
508   \ifx\ZREF@P\ZREF@par
509     \@PackageError\ZREF@name{%
510       Invalid property name `\'ZREF@P'
511     }{%
512       The property name `par' is not allowed %
513       because of internal reasons.%
514       \MessageBreak
515       \@ehc
516     }%
517   \def\ZREF@@newprop[##1]##2{\endgroup}%
518   \else
519     \zref@ifpropundefined\ZREF@P{%
520       \endgroup
521       \PackageInfo\ZREF@name{%
522         New property: \ZREF@P
523       }%
524     }{%
525       \@PackageError\ZREF@name{%
526         Property `\'ZREF@P' already exists%
527       }\@ehc
528     \def\ZREF@@newprop[##1]##2{\endgroup}%
529     }%
530   \fi
531   \@ifnextchar[\ZREF@@newprop{\ZREF@@newprop[\zref@default]}%
532 }
```

(End of definition for \ZREF@newprop.)

\ZREF@par

```

533 \def\ZREF@par{par}
534 \@onelevel@sanitize\ZREF@par
```

(End of definition for \ZREF@par.)

`\ZREF@@newprop`

```

535 \def\ZREF@@newprop[#1]{%
536   \global\@namedef{Z@D@ZREF@P}{#1}%
537   \global\expandafter\let\csname Z@X@ZREF@P\endcsname\ZREF@X
538   \begingroup\expandafter\endgroup
539   \expandafter\ZREF@@@newprop\csname\ZREF@P\endcsname
540   \expandafter\gdef\csname Z@C@ZREF@P\endcsname{}%
541   \zref@setcurrent\ZREF@P
542 }
543 \def\ZREF@@@newprop#1{%
544   \expandafter
545   \gdef\csname Z@E@ZREF@P\endcsname##1#1##2##3\ZREF@nil{##2}%
546 }

```

(End of definition for \ZREF@@newprop.)

`\zref@showprop`

```

547 \ZREF@Robust\def\zref@showprop#1{%
548   \zref@ifpropundefined{#1}{%
549     \@PackageInfoNoLine{\ZREF@name}{%
550       Show property `#1': <undefined>%
551     }%
552   }{%
553     \begingroup
554     \toks@\expandafter\expandafter\expandafter{%
555       \csname Z@C@#1\endcsname
556     }%
557     \edef\ZREF@value{\the\toks@}%
558     \ltx@onelevel@sanitize\ZREF@value
559     \toks@\expandafter\expandafter\expandafter{%
560       \csname Z@D@#1\endcsname
561     }%
562     \edef\ZREF@default{\the\toks@}%
563     \ltx@onelevel@sanitize\ZREF@default
564     \@PackageInfoNoLine{\ZREF@name}{%
565       Show property `#1':\MessageBreak
566       \expandafter\ifx\csname Z@X@#1\endcsname\ltx@empty
567         Immediate %
568       \else
569         Delayed %
570       \fi
571       value: [\ZREF@value]\MessageBreak
572       Default: [\ZREF@default]%
573     }%
574   \endgroup
575 }%
576 }

```

(End of definition for \zref@showprop. This function is documented on page 6.)

`\zref@setcurrent` `\zref@setcurrent` sets the current value for a property.

```

577 \ZREF@Robust\def\zref@setcurrent#1#2{%
578   \zref@propexists{#1}{%
579     \expandafter\def\csname Z@C@#1\endcsname{#2}%
580   }%
581 }

```

(End of definition for \zref@setcurrent. This function is documented on page 7.)

`\ZREF@getcurrent` `\zref@getcurrent` gets the current value for a property.

```

582 \def\ZREF@getcurrent#1{%
583   \romannumeral0%
584   \ltx@ifundefined{Z@C@#1}{%

```

```

585     \ltx@space
586   }{%
587     \expandafter\expandafter\expandafter\ltx@space
588     \csname Z@C@#1\endcsname
589   }%
590 }

```

(End of definition for \ZREF@getcurrent.)

\ZREF@u@getcurrent

```

591 \def\ZREF@u@getcurrent#1{%
592   \etex@unexpanded\expandafter\expandafter\expandafter{%
593     \ZREF@getcurrent{#1}%
594   }%
595 }

```

(End of definition for \ZREF@u@getcurrent.)

\zref@getcurrent

```

596 \let\zref@getcurrent\ZREF@getcurrent

```

(End of definition for \zref@getcurrent. This function is documented on page 7.)

6.2.8 Reference generation

\zref@label Label macro that uses the main property list.

```

597 \ZREF@Robust\def\zref@label#1{%
598   \zref@labelbylist{#1}\ZREF@mainlist
599 }

```

(End of definition for \zref@label. This function is documented on page 7.)

\zref@labelbylist Label macro that stores the properties, specified in the property list #2.

```

600 \ZREF@Robust\def\zref@labelbylist#1#2{%
601   \@bsphack
602   \zref@listexists{#2}{%
603     \expandafter\expandafter\expandafter\ZREF@label
604     \expandafter\expandafter\expandafter{%
605       \csname Z@L@#2\endcsname
606     }{#1}%
607   }%
608   \@esphack
609 }

```

(End of definition for \zref@labelbylist. This function is documented on page 7.)

\zref@labelbyprops The properties are directly specified in a comma separated list.

```

610 \ZREF@Robust\def\zref@labelbyprops#1#2{%
611   \@bsphack
612   \begingroup
613   \toks@{}%
614   \comma@parse{#2}{%
615     \zref@ifpropundefined\comma@entry{%
616       \PackageWarning\ZREF@name{%
617         Property '\comma@entry' is not known%
618       }%
619     }{%
620       \toks@\expandafter{%
621         \the\expandafter\toks@\csname\comma@entry\endcsname
622       }%
623     }%
624     \ltx@gobble
625   }%
626   \expandafter\endgroup

```

```

627     \expandafter\ZREF@label\expandafter{\the\toks@}{#1}%
628     \@esphack
629 }

```

(End of definition for \zref@labelbyprops. This function is documented on page 8.)

\zref@labelbykv

```

630 \ZREF@Robust\def\zref@labelbykv#1#2{%
631     \@bsphack
632     \begingroup
633     \let\Z@L@ZREF@temp\ltx@empty
634     \kvsetkeys{ZREF@LABEL}{#1}%
635     \ifZREF@immediate
636         \expandafter\zref@wrapper@immediate\expandafter{%
637             \expandafter\ZREF@label\expandafter{\Z@L@ZREF@temp}{#2}%
638         }%
639     \else
640         \expandafter\ZREF@label\expandafter{\Z@L@ZREF@temp}{#2}%
641     \fi
642     \endgroup
643     \@esphack
644 }

```

(End of definition for \zref@labelbykv. This function is documented on page 8.)

```

645 \kv@define@key{ZREF@LABEL}{prop}{%
646     \edef\ZREF@param{#1}%
647     \zref@propexists\ZREF@param{%
648         \zref@iflistcontainsprop{ZREF@temp}\ZREF@param{}{%
649             \begingroup\expandafter\endgroup
650             \expandafter\ltx@LocalAppendToMacro
651             \expandafter\Z@L@ZREF@temp
652             \expandafter{\csname\ZREF@param\endcsname}%
653         }%
654     }%
655 }
656 \kv@define@key{ZREF@LABEL}{list}{%
657     \zref@listforloop{#1}{%
658         \zref@iflistcontainsprop{ZREF@temp}\zref@prop{}{%
659             \begingroup\expandafter\endgroup
660             \expandafter\ltx@LocalAppendToMacro
661             \expandafter\Z@L@ZREF@temp
662             \expandafter{\csname\zref@prop\endcsname}%
663         }%
664         \ltx@gobble
665     }%
666 }
667 \kv@define@key{ZREF@LABEL}{delprop}{%
668     \zref@propexists{#1}{%
669         \zref@localdelprop{ZREF@temp}{#1}%
670     }%
671 }
672 \kv@define@key{ZREF@LABEL}{immediate}[true]{%
673     \edef\ZREF@param{#1}%
674     \ifx\ZREF@param\ZREF@true
675         \ZREF@immediatetrue
676     \else
677         \ifx\ZREF@param\ZREF@false
678             \ZREF@immediatefalse
679         \else
680             \PackageWarning\ZREF@name{%
681                 Option `immediate' expects `true' or `false'.\MessageBreak
682                 Ignoring invalid value `\'ZREF@param'%
683             }%

```

```

684     \fi
685   \fi
686 }

```

`\ZREF@false`

```

687 \def\ZREF@false{false}

```

(End of definition for \ZREF@false.)

`\ZREF@true`

```

688 \def\ZREF@true{true}

```

(End of definition for \ZREF@true.)

```

689 \kv@define@key{ZREF@LABEL}{values}[]{%
690   \kv@parse{#1}{%
691     \ifx\kv@value\relax
692       \@PackageWarning\ZREF@name{%
693         Missing value for property `kv@key'%
694       }%
695       \expandafter\ltx@gobbletwo
696     \else
697       \expandafter\zref@setcurrent
698     \fi
699   }%
700 }

```

`\ifZREF@immediate` The switch `\ifZREF@immediate` tells us, whether the label should be written immediately or at page shipout time. `\ZREF@label` need to be notified about this, because it must disable the deferred execution of property values, if the label is written immediately.

```

701 \newif\ifZREF@immediate

```

(End of definition for \ifZREF@immediate.)

`\zref@wrapper@immediate` The argument of `\zref@wrapper@immediate` is executed inside a group where `\write` is redefined by adding `\immediate` before its execution. Also `\ZREF@label` is notified via the switch `\ifZREF@immediate`.

```

702 \ZREF@Robust{\long\def}\zref@wrapper@immediate#1{%
703   \begingroup
704     \ZREF@immediatetrue
705     \let\ZREF@org@write\write
706     \def\write{\immediate\ZREF@org@write}%
707     #1%
708   \endgroup
709 }

```

(End of definition for \zref@wrapper@immediate. This function is documented on page 11.)

`\ZREF@label` `\ZREF@label` writes the data in the `.aux` file. #1 contains the list of valid properties, #2 the name of the reference. In case of immediate writing, the deferred execution of property values is disabled. Also `\ZREF@label` is made expandable in this case.

```

710 \def\ZREF@label#1#2{%
711   \if@filesw
712     \begingroup
713       \ifZREF@immediate
714         \let\ZREF@org@thepage\thepage
715       \fi
716       \protected@write\@auxout{%
717         \let\zlabel\@gobble
718         \let\label\@gobble@om
719         \let\index\@gobble@som
720         \let\glossary\@gobble@om
721         \ifZREF@immediate

```

```

722     \let\thepage\ZREF@org@thepage
723   \fi
724   \let\ZREF@temp\ltx@empty
725   \@tfor\ZREF@P:=#1\do{%
726     \begingroup
727     \escapechar=-1 %
728     \edef\x{\endgroup
729       \def\noexpand\ZREF@P{%
730         \expandafter\string\ZREF@P
731       }%
732     }%
733   \x
734   \expandafter\ifx
735     \csname
736       \ifZREF@immediate
737         relax%
738       \else
739         Z@X@\ZREF@P%
740       \fi
741     \endcsname
742     \noexpand
743     \expandafter\let\csname Z@C@\ZREF@P\endcsname\relax
744   \fi
745   \toks@\expandafter{\ZREF@temp}%
746   \edef\ZREF@temp{%
747     \the\toks@
748     \ltx@backslashchar\ZREF@P%
749     \expandafter\noexpand\csname Z@C@\ZREF@P\endcsname
750   }%
751 }%
752 }%
753 }{%
754   \string\zref@newlabel{#2}{\ZREF@temp}%
755 }%
756 \endgroup
757 \fi
758 }
759 \def\ZREF@addtoks#1{%
760   \toks@\expandafter\expandafter\expandafter{%
761     \expandafter\the\expandafter\toks@#1%
762   }%
763 }

```

(End of definition for \ZREF@label.)

6.2.9 Reference querying and extracting

Design goal for the extracting macros is that the extraction process is full expandable. Thus these macros can be used in expandable contexts. But there are problems that cannot be solved by full expandable macros:

- In standard L^AT_EX undefined references sets a flag and generate a warning. Both actions are not expandable.
- Babel's support for its shorthand uses commands that use non-expandable assignments. However currently there is hope, that primitives are added to pdfT_EX that allows the detection of contexts. Then the shorthand can detect, if they are executed inside \csname and protect themselves automatically.

\zref@ifrefundefined If a reference #1 is undefined, then macro \zref@ifrefundefined calls #2 and #3 otherwise.

```

764 \def\zref@ifrefundefined#1{%
765   \ltx@ifundefined{Z@R@#1}%

```

766 }

(End of definition for \zref@ifrefundefined. This function is documented on page 9.)

\zifrefundefined If a reference #1 is undefined, then macro \zref@ifrefundefined calls #2 and #3 otherwise. Also the reference is marked used.

```
767 \ZREF@ifDefinable\zifrefundefined\def{%
768   #1{%
769     \zref@wrapper@babel\ZREF@ifrefundefined{#1}%
770   }%
771 }
```

(End of definition for \zifrefundefined. This function is documented on page 9.)

\ZREF@ifrefundefined

```
772 \def\ZREF@ifrefundefined#1{%
773   \zref@refused{#1}%
774   \zref@ifrefundefined{#1}%
775 }
```

(End of definition for \ZREF@ifrefundefined.)

\zref@refused The problem with undefined references is addressed by the macro \zref@refused. This can be used outside the expandable context. In case of an undefined reference the flag is set to notify L^AT_EX and a warning is given.

```
776 \ZREF@Robust\def\zref@refused#1{%
777   \zref@wrapper@babel\ZREF@refused{#1}%
778 }
```

(End of definition for \zref@refused. This function is documented on page 8.)

\ZREF@refused

```
779 \def\ZREF@refused#1{%
780   \zref@ifrefundefined{#1}{%
781     \protect\G@refundefinedtrue
782     \@latex@warning{%
783       Reference `#1' on page \thepage \space undefined%
784     }%
785   }{}%
786 }
```

(End of definition for \ZREF@refused.)

\zref@ifrefcontainsprop \zref@ifrefcontainsprop looks, if the reference #1 has the property #2 and calls then #3 and #4 otherwise. #2 must be a defined property.

```
787 \def\zref@ifrefcontainsprop#1#2{%
788   \zref@ifrefundefined{#1}{%
789     \ltx@secondoftwo
790   }{%
791     \expandafter\ZREF@ifrefcontainsprop
792     \csname Z@E@#2\expandafter\endcsname
793     \csname#2\expandafter\expandafter\expandafter\endcsname
794     \expandafter\expandafter\expandafter{%
795       \csname Z@R@#1\endcsname
796     }%
797   }%
798 }
799 \def\ZREF@ifrefcontainsprop#1#2#3{%
800   \expandafter\ifx\expandafter\ZREF@novalue
801   #1#3#2\ZREF@novalue\ZREF@nil\ltx@empty
802   \expandafter\ltx@secondoftwo
803   \else
804   \expandafter\ltx@firstoftwo
805   \fi
806 }
807 \def\ZREF@novalue{\ZREF@NOVALUE}
```

(End of definition for `\zref@ifrefcontainsprop`. This function is documented on page 9.)

`\zref@extract` `\zref@extract` is an abbreviation for the case that the default of the property is used as default value.

```

808 \def\ZREF@extract#1#2{%
809   \romannumeral0%
810   \ltx@ifundefined{Z@D@#2}{%
811     \expandafter\ltx@space\zref@default
812   }{%
813     \expandafter\expandafter\expandafter\ZREF@@extract
814     \expandafter\expandafter\expandafter{%
815       \csname Z@D@#2\endcsname
816     }{#1}{#2}%
817   }%
818 }
```

(End of definition for `\zref@extract`. This function is documented on page 8.)

`\ZREF@@extract`

```

819 \def\ZREF@@extract#1#2#3{%
820   \expandafter\expandafter\expandafter\ltx@space
821   \zref@extractdefault{#2}{#3}{#1}%
822 }
```

(End of definition for `\ZREF@@extract`.)

`\ZREF@wu@extract`

```

823 \def\ZREF@wu@extract#1#2{%
824   \etex@unexpanded\expandafter\expandafter\expandafter{%
825     \ZREF@extract{#1}{#2}%
826   }%
827 }
```

(End of definition for `\ZREF@wu@extract`.)

`\zref@extract`

```

828 \let\zref@extract\ZREF@extract
```

(End of definition for `\zref@extract`. This function is documented on page 8.)

`\ZREF@extractdefault` The basic extracting macro is `\zref@extractdefault` with the reference name in #1, the property in #2 and the default value in #3 in case for problems.

```

829 \def\ZREF@extractdefault#1#2#3{%
830   \romannumeral0%
831   \zref@ifrefundefined{#1}\ltx@firstoftwo{%
832     \zref@ifpropundefined{#2}\ltx@firstoftwo\ltx@secondoftwo
833   }{%
834     \ltx@space
835     #3%
836   }{%
837     \expandafter\expandafter\expandafter\ltx@space
838     \csname Z@E@#2\expandafter\expandafter\expandafter\endcsname
839     \csname Z@R@#1\expandafter\endcsname
840     \csname#2\endcsname{#3}\ZREF@nil
841   }%
842 }
```

(End of definition for `\ZREF@extractdefault`.)

`\ZREF@wu@extractdefault`

```

843 \def\ZREF@wu@extractdefault#1#2#3{%
844   \etex@unexpanded\expandafter\expandafter\expandafter{%
845     \ZREF@extractdefault{#1}{#2}{#3}%
846   }%
847 }
```

(End of definition for \ZREF@wu@extractdefault.)

\zref@extractdefault

```
848 \let\zref@extractdefault\ZREF@extractdefault
```

(End of definition for \zref@extractdefault. This function is documented on page 8.)

\zref@def@extract

```
849 \ZREF@Robust\def\zref@def@extract#1{%  
850   \zref@wrapper@babel{\ZREF@def@extract{#1}}%  
851 }
```

(End of definition for \zref@def@extract. This function is documented on page 8.)

\ZREF@def@extract

```
852 \def\ZREF@def@extract#1#2#3{%  
853   \zref@refused{#2}%  
854   \expandafter\expandafter\expandafter\def  
855   \expandafter\expandafter\expandafter#1%  
856   \expandafter\expandafter\expandafter{%  
857     \zref@extract{#2}{#3}%  
858   }%  
859 }
```

(End of definition for \ZREF@def@extract.)

\zref@def@extractdefault

```
860 \ZREF@Robust\def\zref@def@extractdefault#1{%  
861   \zref@wrapper@babel{\ZREF@def@extractdefault{#1}}%  
862 }
```

(End of definition for \zref@def@extractdefault. This function is documented on page 8.)

\ZREF@def@extractdefault

```
863 \def\ZREF@def@extractdefault#1#2#3#4{%  
864   \zref@refused{#2}%  
865   \expandafter\expandafter\expandafter\def  
866   \expandafter\expandafter\expandafter#1%  
867   \expandafter\expandafter\expandafter{%  
868     \zref@extractdefault{#2}{#3}{#4}%  
869   }%  
870 }
```

(End of definition for \ZREF@def@extractdefault.)

\ZREF@wrapper@unexpanded

```
871 \ZREF@Robust{\long\def}\ZREF@wrapper@unexpanded#1{%  
872   \let\zref@wrapper@unexpanded\ltx@iffirstofone  
873   \let\zref@getcurrent\ZREF@wu@getcurrent  
874   \let\zref@extractdefault\ZREF@wu@extractdefault  
875   \let\zref@extract\ZREF@wu@extract  
876   #1%  
877   \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded  
878   \let\zref@getcurrent\ZREF@getcurrent  
879   \let\zref@extractdefault\ZREF@extractdefault  
880   \let\zref@extract\ZREF@extract  
881 }
```

(End of definition for \ZREF@wrapper@unexpanded.)

\zref@wrapper@unexpanded

```
882 \ltx@ifundefined{etex@unexpanded}{%  
883   \let\zref@wrapper@unexpanded\ltx@iffirstofone  
884 }{%  
885   \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded  
886 }
```

(End of definition for \zref@wrapper@unexpanded. This function is documented on page 11.)

6.2.10 Compatibility with babel

`\zref@wrapper@babel`

```

887 \ZREF@Robust{\long\def}\zref@wrapper@babel#1#2{%
888   \ifcsname if@safe@actives\endcsname
889     \expandafter\ltx@firstofone
890   \else
891     \expandafter\ltx@secondoftwo
892   \fi
893   {%
894     \if@safe@actives
895       \expandafter\ltx@secondoftwo
896     \else
897       \expandafter\ltx@firstoftwo
898     \fi
899     {%
900       \begingroup
901       \csname @safe@activestrue\endcsname
902       \edef\x{#2}%
903       \expandafter\endgroup
904       \expandafter\ZREF@wrapper@babel\expandafter{\x}{#1}%
905     }%
906   }{%
907     #1{#2}%
908   }%
909 }
910 \long\def\ZREF@wrapper@babel#1#2{%
911   #2{#1}%
912 }
```

(End of definition for `\zref@wrapper@babel`. This function is documented on page 10.)

6.2.11 Unique counter support

`\zref@require@unique`

Generate the counter `zref@unique` if the counter does not already exist. change 2022-08-03: The counter is no longer removed from the reset, so that the behaviour with `includeonly` is improved, see issue <https://github.com/ho-tex/zref/issues/10>

```

913 \ZREF@Robust\def\zref@require@unique{%
914   \@ifundefined{c@zref@unique}{%
915     \newcounter{zref@unique}%
916   }
```

(End of definition for `\zref@require@unique`. This function is documented on page 11.)

`\thezref@unique`

`\thezref@unique` is used for automatically generated unique labelnames.

```

916   \renewcommand*{\thezref@unique}{%
917     zref@number\c@zref@unique
918   }%
```

To improve the behaviour if `include/includeonly` is used, we round the counter up at every include, see <https://github.com/ho-tex/zref/issues/10>

```

919   \IfFormatAtLeastTF{2020/10/01}
920   {%
921     \AddToHook
922     {include/after}
923     {\setcounter{zref@unique}
924      {\csname fp_to_int:n\endcsname{ceil(\c@zref@unique +100,-2)}}}
925   }{}%
926 }{}%
927 }
```

(End of definition for `\thezref@unique`. This function is documented on page 11.)

6.2.12 Utilities

`\ZREF@number`

```

928 \ltx@ifundefined{numexpr}{%
929   \def\ZREF@number#1{\number#1}%
930 }{%
931   \def\ZREF@number#1{\the\numexpr(#1)\relax}%
932 }
```

(End of definition for \ZREF@number.)

6.2.13 Setup

`\zref@setdefault` Standard L^AT_EX prints “??” in bold face if a reference is not known. `\zref@default` holds the text that is printed in case of unknown references and is used, if the default was not specified during the definition of the new property by `\ref@newprop`. The global default value can be set by `\zref@setdefault`.

```

933 \ZREF@Robust\def\zref@setdefault#1{%
934   \def\zref@default{#1}%
935 }
```

(End of definition for \zref@setdefault. This function is documented on page 9.)

`\zref@default` Now we initialize `\zref@default` with the same value that L^AT_EX uses for its undefined references.

```

936 \zref@setdefault{%
937   \nfss@text{\reset@font\bfseries ??}%
938 }
```

(End of definition for \zref@default. This function is documented on page 9.)

Main property list.

`\zref@setmainlist` The name of the default property list is stored in `\ZREF@mainlist` and can be set by `\zref@setmainlist`.

```

939 \ZREF@Robust\def\zref@setmainlist#1{%
940   \def\ZREF@mainlist{#1}%
941 }
942 \zref@setmainlist{main}
```

(End of definition for \zref@setmainlist. This function is documented on page 9.)

Now we create the list.

```

943 \zref@newlist\ZREF@mainlist
```

Main properties. The two properties `default` and `page` are created and added to the main property list. They store the data that standard L^AT_EX uses in its references created by `\label`.

default the appearance of the latest counter that is incremented by `\refstepcounter`

page the appearance of the page counter

```

944 \zref@newprop{default}{\@currentlabel}
945 \zref@newprop*{page}{\thepage}
946 \zref@addprops\ZREF@mainlist{default,page}
```

Properties

`\ZREF@NewPropAnchor`

```

947 \def\ZREF@NewPropAnchor{%
948   \zref@newprop{anchor}{}%
949   \ltx@ifundefined{@currentHref}{\@currentHref}%
950 }%
951 \global\let\ZREF@NewPropAnchor\relax
952 }
```

(End of definition for \ZREF@NewPropAnchor.)

`\zref@titleref@current` Later we will redefine the section and caption macros to catch the current title and remember the value in `\zref@titleref@current`.

```

953 \def\ZREF@NewPropTitle{\ZREF@NewPropTitle
954   \IfDocumentMetadataTF
955   {\zref@newprop{title}{\@currentlabelname}}
956   {%
957     \gdef\zref@titleref@current{%
958       \zref@newprop{title}{\zref@titleref@current}%
959     }%
960   \global\let\ZREF@NewPropTitle\relax
961 }
```

(End of definition for \zref@titleref@current and \ZREF@NewPropTitle.)

`\ZREF@NewPropTheotype`

```

962 \def\ZREF@NewPropTheotype{%
963   \zref@newprop{theotype}{}%
964   \global\let\ZREF@NewPropTheotype\relax
965 }
```

(End of definition for \ZREF@NewPropTheotype.)

`\ZREF@NewPropPageValue`

```

966 \def\ZREF@NewPropPageValue{%
967   \zref@newprop*{pagevalue}[0]{\number\c@page}%
968   \global\let\ZREF@NewPropPageValue\relax
969 }
```

(End of definition for \ZREF@NewPropPageValue.)

Mark successful loading

```

970 \let\ZREF@base@ok=Y
971 \</base>
```

6.3 Module user

```

972 <*user>
973 \NeedsTeXFormat{LaTeX2e}
974 \ProvidesPackage{zref-user}%
975   [2026-09-16 v2.38 Module user for zref (H0)]%
976 \RequirePackage{zref-base}[2019/11/29]
977 \ifx\ZREF@base@ok Y%
978 \else
979   \expandafter\endinput
980 \fi
```

Module `user` enables a small user interface. All macros are prefixed by `\z`.

First we define the pendants to the standard L^AT_EX referencing commands `\label`, `\ref`, and `\pageref`.

\zlabel Similar to `\label` the macro `\zlabel` writes a reference entry in the `.aux` file. The main property list is used. Also we add the babel patch. The `\label` command can also be used inside section titles, but it must not go into the table of contents. Therefore we have to check this situation.

```

981 \IfFormatAtLeastTF{2025-06-01}
982 {
983   \newcommand*\zlabel{%
984     \ifx\label\@gobble@om
985       \expandafter\ltx@gobble
986     \else
987       \expandafter\zref@wrapper@babel\expandafter\zref@label
988     \fi
989   }
990 }
991 {
992   \newcommand*\zlabel{%
993     \ifx\label\ltx@gobble
994       \expandafter\ltx@gobble
995     \else
996       \expandafter\zref@wrapper@babel\expandafter\zref@label
997     \fi
998   }
999 }%
```

(End of definition for `\zlabel`. This function is documented on page 12.)

\zkvlabel

```

1000 \newcommand*\zkvlabel}[1]{%
1001   \ifx\label\ltx@gobble
1002     \expandafter\ltx@gobblethree
1003   \fi
1004   \zref@wrapper@babel{\zref@labelbykv{#1}}%
1005 }%
```

(End of definition for `\zkvlabel`. This function is documented on page 12.)

\zref Macro `\zref` is the corresponding macro for `\ref`. Also it provides an optional argument in order to select another property.

```

1006 \newcommand*\zref}[2][default]{% robust because of optional argument
1007   \zref@propexists{#1}{%
1008     \zref@wrapper@babel\ZREF@zref{#2}{#1}%
1009   }%
1010 }%
1011 \def\ZREF@zref#1{%
1012   \zref@refused{#1}%
1013   \zref@extract{#1}%
1014 }%
```

(End of definition for `\zref`. This function is documented on page 12.)

\zpageref For macro `\zpageref` we just call `\zref` with property `page`.

```

1015 \ZREF@IfDefinable\zpageref\def{%
1016   {\zref[page]}%
1017 }
```

(End of definition for `\zpageref`. This function is documented on page 12.)

\zrefused For the following expandable user macros `\zrefused` should be used to notify L^AT_EX in case of undefined references.

```

1018 \ZREF@IfDefinable\zrefused\def{%
1019   {\zref@refused}%
1020 }
```

(End of definition for `\zrefused`. This function is documented on page 12.)

```

1021 \</user>
```

6.4 Module `abspage`

```
1022 <*abspage>
1023 \NeedsTeXFormat{LaTeX2e}
1024 \ProvidesPackage{zref-abspage}%
1025 [2026-09-16 v2.38 Module abspage for zref (H0)]%
1026 \RequirePackage{zref-base}[2019/11/29]
1027 \ifx\ZREF@base@ok Y%
1028 \else
1029 \expandafter\endinput
1030 \fi
```

Module `abspage` adds a new property `abspage` to the main property list for absolute page numbers. These are recorded by the help of the standard LaTeX shipout hooks. The count used as property is the standard LaTeX counter `\ReadonlyShipoutCounter`, but we still provide also the counter `abspage` for compatibility with older code. The counter `abspage` must not go in the clear list of `@ckpt` that is used to set counters in `.aux` files of included `TeX` files.

```
1031 \begingroup
1032 \ifundefined{c@abspage}
1033 {%
1034 \let\@addtoreset\ltx@gobbletwo
1035 \newcounter{abspage}%
1036 }{}
1037 \endgroup
1038 \setcounter{abspage}{0}%
1039 \AddToHook{shipout/before}{%
1040 \stepcounter{abspage}%
1041 }%
1042 \zref@newprop*{abspage}[0]{\the\ReadonlyShipoutCounter}%
1043 \zref@addprop\ZREF@mainlist{abspage}%
```

Note that counter `\ReadonlyShipoutCounter` and `abspage` show the previous page during page processing. Before shipout the counters are incremented. Thus the property is correctly written with deferred writing. If the counter is written using `\zref@wrapper@immediate`, then the number is too small by one.

```
1044 </abspage>
```

6.5 Module `counter`

```
1045 <*counter>
1046 \NeedsTeXFormat{LaTeX2e}
1047 \ProvidesPackage{zref-counter}%
1048 [2026-09-16 v2.38 Module counter for zref (H0)]%
1049 \RequirePackage{zref-base}[2019/11/29]
1050 \ifx\ZREF@base@ok Y%
1051 \else
1052 \expandafter\endinput
1053 \fi
```

For features such as `hyperref`'s `\autoref` we need the name of the counter. The property `counter` is defined and added to the main property list. Starting with `LATEX 2020-10-01` the proper can use `currentcounter`. In older formats `\refstepcounter` has to be patched but this can fail in some cases, see issue #5.

```
1054 \ifl@t@r\fmtversion{2020-10-01}
1055 {
1056 \zref@newprop{counter}{\@currentcounter}
1057 \zref@addprop\ZREF@mainlist{counter}
1058 }
1059 {
1060 \zref@newprop{counter}{}

```

```

1061 \zref@addprop\ZREF@mainlist{counter}
1062 \AtBeginDocument{%
1063   \ZREF@patch{refstepcounter}{%
1064     \def\refstepcounter#1{%
1065       \zref@setcurrent{counter}{#1}%
1066       \ZREF@org@refstepcounter{#1}%
1067     }%
1068   }%
1069 }
1070 }
1071 </counter>

```

6.6 Module lastpage

```

1072 <{*lastpage}>
1073 \NeedsTeXFormat{LaTeX2e}
1074 \ProvidesPackage{zref-lastpage}%
1075   [2026-09-16 v2.38 Module lastpage for zref (HO)]%
1076 \RequirePackage{zref-base}[2019/11/29]
1077 \RequirePackage{zref-abspage}[2019/11/29]
1078 \ifx\ZREF@base@ok Y%
1079 \else
1080   \expandafter\endinput
1081 \fi

```

The module `lastpage` implements the service of package `lastpage` by setting a reference `LastPage` at the end of the document. If module `abspage` is given, also the absolute page number is available, because the properties of the main property list are used. 2025-06-07: adapted to new format which changes the page at end.

```

1082 \zref@newlist{LastPage}
1083 \@ifl@t@r\fmtversion{2025-06-01}{%
1084   \AddToHook {enddocument/afterlastpage}{%
1085     \if@filesw
1086       \begingroup
1087         \toks@{\expandafter\expandafter\expandafter{%
1088           \expandafter\Z@L@main
1089           \Z@L@LastPage
1090         }}%
1091         \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
1092       \endgroup
1093     \fi
1094   }}
1095 {\AddToHook {enddocument/afterlastpage}{%
1096   \if@filesw
1097     \begingroup
1098       \advance\c@page\m@ne
1099       \toks@{\expandafter\expandafter\expandafter{%
1100         \expandafter\Z@L@main
1101         \Z@L@LastPage
1102       }}%
1103       \expandafter\zref@wrapper@immediate\expandafter{%
1104         \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
1105       }%
1106     \endgroup
1107   \fi
1108 }}

```

`\zref@iflastpage`

```

1109 \def\zref@iflastpage#1{%
1110   \ifnum\zref@extractdefault{#1}{abspage}{-1}=%
1111     \zref@extractdefault{LastPage}{abspage}{-2} %

```

```

1112     \expandafter\ltx@firstoftwo
1113   \else
1114     \expandafter\ltx@secondoftwo
1115   \fi
1116 }

```

(End of definition for \zref@iflastpage. This function is documented on page 13.)

\ziflastpage

```

1117 \ZREF@ifdefinable\ziflastpage\def{%
1118   {\zref@wrapper@babel\ZREF@iflastpage}%
1119 }

```

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

\ZREF@iflastpage

```

1120 \def\ZREF@iflastpage#1{%
1121   \zref@refused{LastPage}%
1122   \zref@refused{#1}%
1123   \zref@iflastpage{#1}%
1124 }

```

(End of definition for \ZREF@iflastpage.)

```

1125 </lastpage>

```

6.7 Module thepage

```

1126 <*thepage>
1127 \NeedsTeXFormat{LaTeX2e}
1128 \ProvidesPackage{zref-thepage}%
1129   [2026-09-16 v2.38 Module thepage for zref (H0)]%
1130 \RequirePackage{zref-base}[2019/11/29]
1131 \ifx\ZREF@base@ok Y%
1132   \else
1133     \expandafter\endinput
1134   \fi

1135 \RequirePackage{zref-abspace}[2019/11/29]

1136 \zref@newlist{thepage}
1137 \zref@addprop{thepage}{page}
1138 \ZREF@NewPropPageValue

```

\zref@thepage@atbegshi@hook

```

1139 \let\zref@thepage@atbegshi@hook\ltx@empty

```

(End of definition for \zref@thepage@atbegshi@hook.)

```

1140 \zref@addprop{thepage}{pagevalue}
1141 \AddToHook{shipout/background}{%
1142   \zref@labelbylist{thepage\the\value{abspace}}{thepage}%
1143 }

```

\zref@thepage@name

```

1144 \ltx@ifundefined{numexpr}{%
1145   \def\zref@thepage@name#1{thepage\number#1}%
1146 }{%
1147   \def\zref@thepage@name#1{thepage\the\numexpr#1}%
1148 }

```

(End of definition for \zref@thepage@name. This function is documented on page 14.)

\zref@thepage

```

1149 \def\zref@thepage#1{%
1150   \zref@extract{\zref@thepage@name{#1}}{page}%
1151 }%

```

(End of definition for \zref@thepage. This function is documented on page 15.)

\zref@thepage@refused

```

1152 \ZREF@Robust\def\zref@thepage@refused#1{%
1153   \zref@refused{\zref@thepage@name{#1}}%
1154 }%

```

(End of definition for \zref@thepage@refused. This function is documented on page 15.)

\zthepage

```

1155 \ZREF@IfDefinable\zthepage\def{%
1156   #1{%
1157     \zref@thepage@refused{#1}%
1158     \zref@thepage{#1}%
1159   }%
1160 }

```

(End of definition for \zthepage. This function is documented on page 14.)

```

1161 \</thepage>

```

6.8 Module nextpage

```

1162 \<*nextpage>
1163 \NeedsTeXFormat{LaTeX2e}
1164 \ProvidesPackage{zref-nextpage}%
1165   [2026-09-16 v2.38 Module nextpage for zref (H0)]%
1166 \RequirePackage{zref-base}[2019/11/29]
1167 \ifx\ZREF@base@ok Y%
1168 \else
1169   \expandafter\endinput
1170 \fi

1171 \RequirePackage{zref-abspage}[2019/11/29]
1172 \RequirePackage{zref-thepage}[2019/11/29]
1173 \RequirePackage{zref-lastpage}[2019/11/29]
1174 \RequirePackage{uniquecounter}[2009/12/18]

1175 \UniqueCounterNew{znnextpage}
1176
1177 \newcommand*{\znnextpagesetup}{%
1178   \afterassignment\ZREF@np@setup@i
1179   \def\ZREF@np@call@unknown##1%
1180 }
1181 \def\ZREF@np@setup@i{%
1182   \afterassignment\ZREF@np@setup@ii
1183   \def\ZREF@np@call@nonext##1%
1184 }
1185 \def\ZREF@np@setup@ii{%
1186   \def\ZREF@np@call@next##1%
1187 }
1188 \def\ZREF@np@call@unknown#1{#1}
1189 \def\ZREF@np@call@nonext#1{#1}
1190 \def\ZREF@np@call@next#1{#1}
1191 \ZREF@IfDefinable\znnextpage\def{%
1192   {\UniqueCounterCall{znnextpage}{\ZREF@nextpage}}%
1193 }%
1194 \newcommand*{\znonextpagename}{\Z@D@page}
1195 \newcommand*{\zunknownnextpagename}{\Z@D@page}

```

```

1196 \def\ZREF@nextpage#1{%
1197   \begingroup
1198     \def\ZREF@refname@this{zref@np#1}%
1199     \zref@labelbyprops\ZREF@refname@this{abspage}%
1200     \chardef\ZREF@call=0 % unknown
1201     \ZREF@ifrefundefined\ZREF@refname@this{%
1202       {%
1203         \edef\ZREF@pagenum@this{%
1204           \zref@extractdefault\ZREF@refname@this{abspage}{0}%
1205         }%
1206         \edef\ZREF@refname@next{%
1207           \zref@thepage@name{%
1208             \the\numexpr\ZREF@pagenum@this+1%
1209           }%
1210         }%
1211         \ifnum\ZREF@pagenum@this>0 %
1212           \ZREF@ifrefundefined{LastPage}{%
1213             \zref@ifrefundefined\ZREF@refname@next{%
1214               {%
1215                 \chardef\ZREF@call=2 % next page
1216               }%
1217             }{%
1218               \edef\ZREF@pagenum@last{%
1219                 \zref@extractdefault{LastPage}{abspage}{0}%
1220               }%
1221               \ifnum\ZREF@pagenum@this<\ZREF@pagenum@last\ltx@space
1222                 \ZREF@ifrefundefined\ZREF@refname@next{%
1223                   {%
1224                     \chardef\ZREF@call=2 % next page
1225                   }%
1226                 }%
1227                 \ifnum\ZREF@pagenum@this=\ZREF@pagenum@last\ltx@space
1228                   \chardef\ZREF@call=1 % no next page
1229                 \fi
1230               }%
1231             }%
1232           }%
1233         }%
1234       \edef\x{%
1235         \endgroup
1236         \ifcase\ZREF@call
1237           \noexpand\ZREF@np@call@unknown{%
1238             \noexpand\zunknownnextpagename
1239           }%
1240         \or
1241           \noexpand\ZREF@np@call@nonext{%
1242             \noexpand\znonextpagename
1243           }%
1244         \else
1245           \noexpand\ZREF@np@call@next{%
1246             \noexpand\zref@extract{\ZREF@refname@next}{page}%
1247           }%
1248         \fi
1249       }%
1250     \x
1251   }
1252 </nextpage>

```

6.9 Module totpages

```

1253 <*totpages>
1254 \NeedsTeXFormat{LaTeX2e}

```

```

1255 \ProvidesPackage{zref-totpages}%
1256 [2026-09-16 v2.38 Module totpages for zref (H0)]%
1257 \RequirePackage{zref-base}[2019/11/29]
1258 \ifx\ZREF@base@ok Y%
1259 \else
1260 \expandafter\endinput
1261 \fi

```

The absolute page number of the last page is the total page number.

```

1262 \RequirePackage{zref-abspace}[2019/11/29]
1263 \RequirePackage{zref-lastpage}[2019/11/29]

```

\ztotpages Macro `\ztotpages` contains the number of pages. It can be used inside expandable calculations. It expands to zero if the reference is not yet available.

```

1264 \newcommand*{\ztotpages}{%
1265   \zref@extractdefault{LastPage}{abspace}{0}%
1266 }

```

(End of definition for `\ztotpages`. This function is documented on page 16.)

Also we mark the reference `LastPage` as used:

```

1267 \AtBeginDocument{%
1268   \zref@refused{LastPage}%
1269 }

```

```

1270 \</totpages>

```

6.10 Module `pagelayout`

```

1271 \<*pagelayout>
1272 \NeedsTeXFormat{LaTeX2e}
1273 \ProvidesPackage{zref-pagelayout}%
1274 [2026-09-16 v2.38 Module pagelayout for zref (H0)]%
1275 \RequirePackage{zref-base}[2019/11/29]
1276 \ifx\ZREF@base@ok Y%
1277 \else
1278 \expandafter\endinput
1279 \fi

```

```

1280 \RequirePackage{zref-thepage}[2019/11/29]
1281 \RequirePackage{iftex}[2019/11/07]%

```

6.10.1 Define layout properties

```

1282 \def\ZREF@temp#1{%
1283   \begingroup
1284     \escapechar=-1 %
1285     \ltx@ifundefined{\string#1}{\endgroup}{%
1286       \edef\x{%
1287         \endgroup
1288         \noexpand\zref@newprop*{\string#1}%
1289                               [\noexpand\number\noexpand#1]% hash-ok
1290                               {\noexpand\number\noexpand#1}%
1291         \noexpand\zref@addprop{thepage}{\string#1}%
1292       }%
1293       \x
1294     }%
1295   }
1296   \ZREF@temp\mag
1297   \ZREF@temp\paperwidth
1298   \ZREF@temp\paperheight
1299   \ZREF@temp\stockwidth % memoir.cls, crop.sty
1300   \ZREF@temp\stockheight % memoir.cls, crop.sty

```

```

1301 \ZREF@temp\mediawidth % VTeX
1302 \ZREF@temp\mediaheight % VTeX
1303 \ifluatex
1304 \zref@newprop*{pdfvorigin}%
1305 [\number\pdfvariable vorigin]% hash-ok
1306 {\number\pdfvariable vorigin}%
1307 \zref@addprop{thepage}{pdfvorigin}
1308 \zref@newprop*{pdfhorigin}%
1309 [\number\pdfvariable horigin]% hash-ok
1310 {\number\pdfvariable horigin}%
1311 \zref@addprop{thepage}{pdfhorigin}
1312 \zref@newprop*{pdfpageheight}%
1313 [\number\pageheight]% hash-ok
1314 {\number\pageheight}%
1315 \zref@addprop{thepage}{pdfpageheight}
1316 \zref@newprop*{pdfpagewidth}%
1317 [\number\pagewidth]% hash-ok
1318 {\number\pagewidth}%
1319 \zref@addprop{thepage}{pdfpagewidth}
1320 \else
1321 \ZREF@temp\pdfpagewidth
1322 \ZREF@temp\pdfpageheight
1323 \ZREF@temp\pdfhorigin
1324 \ZREF@temp\pdfvorigin
1325 \fi
1326 \ZREF@temp\hoffset
1327 \ZREF@temp\voffset
1328 \ZREF@temp\topmargin
1329 \ZREF@temp\oddsidemargin
1330 \ZREF@temp\evensidemargin
1331 \ZREF@temp\textwidth
1332 \ZREF@temp\textheight
1333 \ZREF@temp\headheight
1334 \ZREF@temp\headsep
1335 \ZREF@temp\footskip
1336 \ZREF@temp\marginparwidth
1337 \ZREF@temp\marginparsep
1338 \ZREF@temp\columnwidth
1339 \ZREF@temp\columnsep
1340 \ZREF@temp\trimedge % memoir.cls
1341 \ZREF@temp\spinemargin % memoir.cls
1342 \ZREF@temp\foremargin % memoir.cls
1343 \ZREF@temp\trintop % memoir.cls
1344 \ZREF@temp\uppermargin % memoir.cls
1345 \ZREF@temp\headmargin % memoir.cls

1346 \ExplSyntaxOn
1347 \zref@newprop*{outputboxwd}[Opt]{\dim_use:N \l_shipout_box_wd_dim}
1348 \zref@newprop*{outputboxht}[Opt]{\dim_use:N \l_shipout_box_ht_dim}
1349 \zref@newprop*{outputboxdp}[Opt]{\dim_use:N \l_shipout_box_dp_dim}
1350 \ExplSyntaxOff

1351 \zref@addprops{thepage}{outputboxwd,outputboxht,outputboxdp}

\ifZREF@pl@list
1352 \ltx@newif\ifZREF@pl@list
(End of definition for \ifZREF@pl@list.)

\zlistpagelayout
1353 \ZREF@IfDefinable\zlistpagelayout\def{%
1354 {\global\ZREF@pl@listtrue}%
1355 }

```

(End of definition for \zlistpagelayout. This function is documented on page 17.)

\ZREF@pl@AfterLastShipout

```

1356 \def\ZREF@pl@AfterLastShipout{%
1357   \ifZREF@pl@list
1358     \edef\ZREF@page@max{\the\value{abspage}}%
1359     \ltx@ifundefined{ZREF@org@testdef}{%
1360       \let\ZREF@org@testdef\@testdef
1361       \def\@testdef##1##2##3{%
1362         \ZREF@org@testdef{##1}{##2}{##3}%
1363       \def\ZREF@temp{##1}%
1364       \ifx\ZREF@temp\ZREF@RefPrefix
1365         \expandafter\gdef\csname##1@##2\endcsname{##3}%
1366       \fi
1367     }%
1368   }{}%
1369   \AddToHook{enddocument/afteraux}{\ZREF@pl@AtVeryEnd}%
1370 \fi
1371 }
```

(End of definition for \ZREF@pl@AfterLastShipout.)

\ZREF@pl@AtVeryEnd

```

1372 \def\ZREF@pl@AtVeryEnd{%
1373   \begingroup
1374   \toks@{\Page layout parameters:\MessageBreak}%
1375   \count@=1 %
1376   \ZREF@pl@ListPage
1377   \edef\x{\endgroup
1378     \noexpand\@PackageInfoNoLine{zref-pagelayout}{\the\toks@}%
1379   }%
1380   \x
1381 }
```

(End of definition for \ZREF@pl@AtVeryEnd.)

\ZREF@pl@ListPage

```

1382 \def\ZREF@pl@ListPage{%
1383   \edef\x{%
1384     \toks@={%
1385       \the\toks@
1386       Page \the\count@:\noexpand\MessageBreak
1387       \zref@ifrefundefined{thepage\the\count@}{}{%
1388         \ltx@space\ltx@space mag = %
1389         \zref@extract{thepage\the\count@}{mag}%
1390         \noexpand\MessageBreak
1391         \ZREF@pl@ListEntry{paperwidth}%
1392         \ZREF@pl@ListEntry{paperheight}%
1393         \ZREF@pl@ListEntry{stockwidth}%
1394         \ZREF@pl@ListEntry{stockheight}%
1395         \ZREF@pl@ListEntry{mediawidth}%
1396         \ZREF@pl@ListEntry{mediaheight}%
1397         \ZREF@pl@ListEntry{pdfpagewidth}%
1398         \ZREF@pl@ListEntry{pdfpageheight}%
1399         \ZREF@pl@ListEntry{pdfhorigin}%
1400         \ZREF@pl@ListEntry{pdfvorigin}%
1401         \ZREF@pl@ListEntry{hoffset}%
1402         \ZREF@pl@ListEntry{voffset}%
1403         \ZREF@pl@ListEntry{topmargin}%
1404         \ZREF@pl@ListEntry{oddsidemargin}%
1405         \ZREF@pl@ListEntry{evensidemargin}%
1406         \ZREF@pl@ListEntry{textwidth}%
1407         \ZREF@pl@ListEntry{textheight}%

```

```

1408      \ZREF@pl@ListEntry{headheight}%
1409      \ZREF@pl@ListEntry{headsep}%
1410      \ZREF@pl@ListEntry{footskip}%
1411      \ZREF@pl@ListEntry{marginparwidth}%
1412      \ZREF@pl@ListEntry{marginparsep}%
1413      \ZREF@pl@ListEntry{columnwidth}%
1414      \ZREF@pl@ListEntry{columnsep}%
1415      \ZREF@pl@ListEntry{trimedge}%
1416      \ZREF@pl@ListEntry{spinemargin}%
1417      \ZREF@pl@ListEntry{foremargin}%
1418      \ZREF@pl@ListEntry{trimtop}%
1419      \ZREF@pl@ListEntry{uppermargin}%
1420      \ZREF@pl@ListEntry{headmargin}%
1421    }%
1422  }%
1423 } \x
1424 \ifnum\ZREF@page@max>\count@
1425   \advance\count@ by\ltx@one
1426 \else
1427   \expandafter\ltx@gobble
1428 \fi
1429 \ZREF@pl@ListPage
1430 }

```

(End of definition for \ZREF@pl@ListPage.)

\ZREF@pl@ListEntry

```

1431 \def\ZREF@pl@ListEntry#1{%
1432   \zref@ifpropundefined{#1}{%
1433     }{%
1434       \zref@ifrefcontainsprop{thepage\the\count@}{#1}{%
1435         \ltx@space\ltx@space#1 = %
1436         \zref@extract{thepage\the\count@}{#1}sp = %
1437         \the\dimexpr\zref@extract{thepage\the\count@}{#1}sp\relax
1438         \noexpand\MessageBreak
1439       }{%
1440       }%
1441     }

```

(End of definition for \ZREF@pl@ListEntry.)

```

1442 \AddToHook {enddocument/afterlastpage}{%
1443   \ZREF@pl@AfterLastShipout
1444 }
1445 </pagelayout>

```

6.11 Module pageattr

```

1446 <*pageattr>
1447 \NeedsTeXFormat{LaTeX2e}
1448 \ProvidesPackage{zref-pageattr}%
1449   [2026-09-16 v2.38 Module pageattr for zref (H0)]%
1450 \RequirePackage{zref-base}[2019/11/29]
1451 \ifx\ZREF@base@ok Y%
1452   \else
1453     \expandafter\endinput
1454   \fi
1455 \RequirePackage{iftex}[2019/11/07]%
1456 \let\ZREF@temp=N%
1457 \ifluatex
1458   \expandafter\@firstoftwo

```

```

1459 \else
1460 \expandafter\@secondoftwo
1461 \fi
1462 {%\luatex
1463 \RequirePackage{zref-thepage}[2019/11/29]
1464 \RequirePackage{zref-lastpage}[2019/11/29]%
1465 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfvariable pageattr}}%
1466 \zref@addprop{thepage}{pdfpageattr}%
1467 \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfvariable pagesattr}}%
1468 \zref@addprop{LastPage}{pdfpagesattr}%
1469 \let\ZREF@temp=Y%
1470 }
1471 {%\other
1472 \ltx@ifundefined{pdfpageattr}{%
1473 \@PackageInfoNoLine{zref-pageattr}{%
1474 \string\pdfpageattr\space is not available%
1475 }%
1476 \def\zref@pdfpageattr#1{%
1477 \def\zref@pdfpageattr@used#1}%
1478 }{%
1479 \RequirePackage{zref-thepage}[2019/11/29]%
1480 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfpageattr}}%
1481 \zref@addprop{thepage}{pdfpageattr}%
1482 \let\ZREF@temp=Y%
1483 }
1484 \ltx@ifundefined{pdfpagesattr}{%
1485 \@PackageInfoNoLine{zref-pageattr}{%
1486 \string\pdfpagesattr\space is not available%
1487 }%
1488 \def\zref@pdfpagesattr{%
1489 \def\zref@pdfpagesattr@used{%
1490 }{%
1491 \RequirePackage{zref-lastpage}[2019/11/29]%
1492 \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfpagesattr}}%
1493 \zref@addprop{LastPage}{pdfpagesattr}%
1494 \let\ZREF@temp=Y%
1495 }%
1496 }%

1497 \ifx\ZREF@temp N%
1498 \expandafter\endinput
1499 \fi

1500 \RequirePackage{zref-abspage}[2019/11/29]
1501 \RequirePackage{pdftexcmds}[2010/04/01]
1502 \let\ZREF@temp=Y%
1503 \ltx@ifundefined{pdf@escapehex}{\let\ZREF@temp=N}{%
1504 \ltx@ifundefined{pdf@unescapehex}{\let\ZREF@temp=N}{%
1505 \ifx\ZREF@temp N%
1506 \let\zref@hex\ltx@firstofone
1507 \let\zref@unhex\ltx@firstofone
1508 \else
1509 \let\zref@hex\pdf@escapehex
1510 \let\zref@unhex\pdf@unescapehex
1511 \fi

```

\ifZREF@pa@list

```
1512 \ltx@newif\ifZREF@pa@list
```

(End of definition for \ifZREF@pa@list.)

\zlistpageattr TODO: perhaps document

```
1513 \ZREF@ifdefinable\zlistpageattr\def{%
```

```

1514     {\ZREF@pa@listtrue}%
1515 }

```

(End of definition for \zlistpageattr.)

\ZREF@pa@AfterLastShipout

```

1516 \def\ZREF@pa@AfterLastShipout{%
1517   \ifZREF@pa@list
1518     \edef\ZREF@page@max{\the\value{abspage}}%
1519     \ltx@ifundefined{ZREF@org@testdef}{%
1520       \let\ZREF@org@testdef\@testdef
1521       \def\@testdef##1##2##3{%
1522         \ZREF@org@testdef{##1}{##2}{##3}%
1523         \def\ZREF@temp{##1}%
1524         \ifx\ZREF@temp\ZREF@RefPrefix
1525           \expandafter\xdef\csname##1@##2\endcsname{##3}%
1526         \fi
1527       }%
1528     }{}%
1529     \AddToHook{enddocument/afteraux}{\ZREF@pa@AtVeryEnd}%
1530   \fi
1531 }

```

(End of definition for \ZREF@pa@AfterLastShipout.)

\ZREF@pa@AtVeryEnd

```

1532 \let\ZREF@temp=Y%
1533 \ltx@ifundefined{pdfpageattr}{}{\let\ZREF@temp=N}
1534 \ifluatex \let\ZREF@temp=N \fi
1535 \ifx\ZREF@temp Y
1536   \expandafter\@firstoftwo
1537 \else
1538   \expandafter\@secondoftwo
1539 \fi
1540 {%
1541   \def\ZREF@pa@AtVeryEnd{%
1542   }
1543   {%
1544     \def\ZREF@pa@AtVeryEnd{%
1545       \begingroup
1546         \toks@{List of \ltx@backslashchar
1547           \ifluatex pdfvariable\else pdf\fi
1548           pdfpageattr:\MessageBreak}%
1549         \count@=1 %
1550         \ZREF@pa@ListPage
1551         \edef\x{\endgroup
1552           \noexpand\@PackageInfoNoLine{zref-pageattr}{%
1553             \the\toks@
1554           }%
1555         }%
1556       \x
1557     }%

```

(End of definition for \ZREF@pa@AtVeryEnd.)

\zref@pdfpageattr

```

1558 \def\zref@pdfpageattr#1{%
1559   \zref@unhex{%
1560     \zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}%
1561   }%
1562 }
1563 % compatibility, \zref@pageattr was defined in older versions
1564 \let\zref@pageattr\zref@pdfpageattr

```

(End of definition for \zref@pdfpageattr. This function is documented on page 22.)

\zref@pageattr@used

```
1565 \ZREF@Robust\def\zref@pageattr@used#1{%
1566   \zref@refused{thepage\ZREF@number{#1}}%
1567 }
```

(End of definition for \zref@pageattr@used.)

\ZREF@pa@ListPage

```
1568 \def\ZREF@pa@ListPage{%
1569   \edef\x{%
1570     \toks@={%
1571       \the\toks@
1572       Page \the\count@:%
1573       \noexpand\MessageBreak
1574       \zref@ifrefundefined{thepage\the\count@}{-}{%
1575         <<\zref@pdfpageattr\count@>>%
1576         \noexpand\MessageBreak
1577       }%
1578     }%
1579   }\x
1580   \ifnum\ZREF@page@max>\count@
1581     \advance\count@ by\ltx@one
1582   \else
1583     \expandafter\ltx@gobble
1584   \fi
1585   \ZREF@pa@ListPage
1586 }%
1587 }
```

(End of definition for \ZREF@pa@ListPage.)

```
1588 \let\ZREF@temp=Y%
1589 \ltx@ifundefined{pdfpagesattr}{-}{\let\ZREF@temp=N}
1590 \ifluatex \let\ZREF@temp=N \fi
1591 \ifx\ZREF@temp N
1592   \expandafter\@firstofone
1593 \fi
1594 {%
```

\zref@pdfpagesattr

```
1595 \def\zref@pdfpagesattr{%
1596   \zref@unhex{%
1597     \zref@extract{LastPage}{pdfpagesattr}%
1598   }%
1599 }
```

(End of definition for \zref@pdfpagesattr. This function is documented on page 22.)

\zref@pdfpagesattr@used

```
1600 \ZREF@Robust\def\zref@pdfpagesattr@used{%
1601   \zref@refused{LastPage}%
1602 }
```

(End of definition for \zref@pdfpagesattr@used. This function is documented on page 22.)

```
1603 \ltx@LocalAppendToMacro\ZREF@pa@AtVeryEnd{%
1604   \@PackageInfoNoLine{zref-pageattr}{%
1605     \ltx@backslashchar
1606     \ifluatex pdfvariable\else pdf\fi
1607     pagesattr:\MessageBreak
1608     <<\zref@pdfpagesattr>>%
1609     \MessageBreak
```

```

1610 }%
1611 }%
1612 }
1613 \AddToHook {enddocument/afterlastpage}{%
1614 \ZREF@pa@AfterLastShipout
1615 }
1616 </pageattr>

```

6.12 Module marks

```

1617 <*marks>
1618 \NeedsTeXFormat{LaTeX2e}
1619 \ProvidesPackage{zref-marks}%
1620 [2026-09-16 v2.38 Module marks for zref (H0)]%
1621 \RequirePackage{zref-base}[2019/11/29]
1622 \ifx\ZREF@base@ok Y%
1623 \else
1624 \expandafter\endinput
1625 \fi

1626 \newcommand*{\zref@marks@register}[3][{}]{%
1627 \edef\ZREF@TempName{#1}%
1628 \edef\ZREF@TempNum{\ZREF@number{#2}}%
1629 \ifnum\ZREF@TempNum<\ltx@zero %
1630 \PackageError\ZREF@name{%
1631 \string\zref@marks@register\ltx@space is called with invalid%
1632 \MessageBreak
1633 marks register number (\ZREF@TempNum)%
1634 }{%
1635 Use `0' or the command, defined by \string\newmarks.\MessageBreak
1636 \@ehc
1637 }%
1638 \else
1639 \ifx\ZREF@TempName\ltx@empty
1640 \edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}%
1641 \else
1642 \edef\ZREF@TempName{marks\ZREF@TempName}%
1643 \fi
1644 \ZREF@MARKS@DefineProp{top}%
1645 \ZREF@MARKS@DefineProp{first}%
1646 \ZREF@MARKS@DefineProp{bot}%
1647 \kv@parse{#3}{%
1648 \ifx\kv@value\relax
1649 \def\kv@value{top,first,bot}%
1650 \fi
1651 \edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}%
1652 \ifx\ZREF@temp\ZREF@STAR
1653 \edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}%
1654 \zref@newlist\kv@key
1655 \fi
1656 \expandafter\comma@parse\expandafter{\kv@value}{%
1657 \ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else
1658 \ifx\comma@entry\ZREF@NAME@first 1\else
1659 \ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space
1660 \PackageWarning{zref-marks}{%
1661 Use `top', `first' or `bot' for the list values%
1662 \MessageBreak
1663 in the third argument of \string\zref@marks@register.%
1664 \MessageBreak
1665 Ignoring unknown value ``\comma@entry'%
1666 }%
1667 \else

```

```

1668         \zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}%
1669     \fi
1670     \ltx@gobble
1671 }%
1672 \ltx@gobbletwo
1673 }%
1674 \fi
1675 }
1676 \def\ZREF@STAR{*}
1677 \def\ZREF@NAME@top{top}
1678 \def\ZREF@NAME@first{first}
1679 \def\ZREF@NAME@bot{bot}
1680 \def\ZREF@MARKS@DefineProp#1{%
1681     \zref@ifpropundefined{#1\ZREF@TempName}{%
1682         \ifnum\ZREF@TempNum=\ltx@zero
1683             \begingroup
1684                 \edef\x{\endgroup
1685                     \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
1686                         \expandafter\noexpand\csname#1mark\endcsname
1687                     }%
1688                 }%
1689                 \x
1690             \else
1691                 \begingroup
1692                     \edef\x{\endgroup
1693                         \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
1694                             \expandafter\noexpand\csname#1marks\endcsname
1695                             \ZREF@TempNum
1696                         }%
1697                     }%
1698                     \x
1699                 \fi
1700             }{%
1701                 \PackageWarning{zref-marks}{%
1702                     \string\zref@marks@register\ltx@space does not generate the%
1703                     \MessageBreak
1704                     new property `#1\ZREF@TempName', because\MessageBreak
1705                     it is already defined%
1706                 }%
1707             }%
1708         }
1709 </marks>

```

6.13 Module runs

This module does not use the label-reference-system. The reference changes with each L^AT_EX run and would force a rerun warning always.

```

1710 <*runs>
1711 \NeedsTeXFormat{LaTeX2e}
1712 \ProvidesPackage{zref-runs}%
1713 [2026-09-16 v2.38 Module runs for zref (H0)]%

```

\zruns

```

1714 \providecommand*{\zruns}{0}%
1715 \AtBeginDocument{%
1716     \edef\zruns{\number\numexpr\zruns+1}%
1717     \begingroup
1718         \def\on@line{%
1719             \PackageInfo{zref-runs}{LaTeX runs: \zruns}%
1720             \if@filesw
1721                 \immediate\write\@mainaux{%

```

```

1722     \string\gdef\string\zruns{\zruns}%
1723   }%
1724   \fi
1725   \endgroup
1726 }

```

(End of definition for `\zruns`. This function is documented on page 17.)

```

1727 </runs>

```

6.14 Module `perpage`

```

1728 <*perpage>
1729 \NeedsTeXFormat{LaTeX2e}
1730 \ProvidesPackage{zref-perpage}%
1731   [2026-09-16 v2.38 Module perpage for zref (HO)]%
1732 \RequirePackage{zref-base}[2019/11/29]
1733 \ifx\ZREF@base@ok Y%
1734 \else
1735   \expandafter\endinput
1736 \fi

```

This module resets a counter at page boundaries. Because of the asynchronous output routine page counter properties cannot be asked directly, references are necessary.

For detecting changed pages module `abspage` is loaded.

```

1737 \RequirePackage{zref-abspage}[2019/11/29]

```

We group the properties for the needed references in the property list `perpage`. The property `pagevalue` records the correct value of the page counter.

```

1738 \ZREF@NewPropPageValue
1739 \zref@newlist{perpage}
1740 \zref@addprops{perpage}{abspage,page,pagevalue}

```

The page value, known by the reference mechanism, will be stored in counter `zpage`.

```

1741 \newcounter{zpage}

```

Counter `zref@unique` helps in generating unique reference names.

```

1742 \zref@require@unique

```

In order to be able to reset the counter, we hook here into `\stepcounter`. In fact two nested hooks are used to allow other packages to use the first hook at the beginning of `\stepcounter`.

```

1743 \let\ZREF@org@stepcounter\stepcounter
1744 \def\stepcounter#1{%
1745   \ifcsname @stepcounterhook@#1\endcsname
1746     \csname @stepcounterhook@#1\endcsname
1747   \fi
1748   \ZREF@org@stepcounter{#1}%
1749 }

```

If `amstext` is loaded it overwrites the definition (or we overwrite their definition) so we account for this by using a package hook, see <https://github.com/ho-tex/zref/issues/11>.

```

1750 \IfFormatAtLeastTF{2020/10/01}
1751 {
1752   \AddToHook{package/amstext/after}
1753   {\def\stepcounter#1{%
1754     \iffirstchoice@
1755     \ifcsname @stepcounterhook@#1\endcsname
1756       \csname @stepcounterhook@#1\endcsname
1757     \fi
1758     \addtocounter{#1}\@ne
1759     \begingroup \let\@elt\@stpel \csname cl@#1\endcsname \endgroup

```

```

1760     \fi
1761   }}
1762 }{}
1763 %   \end{macrocode}
1764 %
1765 %
1766 %   \cs{@stpelt} must be adapted due to the change in latex
1767 %   2015-01, see https://github.com/ho-tex/zref/issues/26
1768 %   \begin{macrocode}
1769 \let\ZREF@org@@stpelt\@stpelt
1770 \def\@stpelt#1{%
1771   \ifcsname ZREF@perpage@#1\endcsname
1772     \begingroup
1773       \let\stepcounter\ZREF@org@stepcounter
1774       \ZREF@org@@stpelt{#1}%
1775     \endgroup
1776     \expandafter\ltx@gobbletwo
1777   \fi
1778   \ZREF@org@@stpelt{#1}%
1779 }

```

\zmakeperpage Macro `\zmakeperpage` resets a counter at each page break. It uses the same syntax and semantics as `\MakePerPage` from package `perpage` [5]. The initial start value can be given by the optional argument. Default is one that means after the first `\stepcounter` on a new page the counter starts with one.

```

1780 \ZREF@ifDefinable\zmakeperpage\def{%
1781   {%
1782     \ifnextchar[\ZREF@makeperpage@opt{\ZREF@@makeperpage[\ltx@zero]}%
1783   }%
1784 }

```

(End of definition for `\zmakeperpage`. This function is documented on page 18.)

We hook before the counter is incremented in `\stepcounter`, package `perpage` afterwards. Thus a little calculation is necessary.

```

1785 \def\ZREF@makeperpage@opt[#1]{%
1786   \begingroup
1787     \edef\x{\endgroup
1788       \noexpand\ZREF@@makeperpage[\number\numexpr#1-1\relax]%
1789     }%
1790     \x
1791 }

1792 \def\ZREF@@makeperpage[#1]#2{%
1793   \ifundefined{@stepcounterhook@#2}{%
1794     \expandafter\gdef\csname @stepcounterhook@#2\endcsname{%
1795   }{}%
1796   \expandafter\gdef\csname ZREF@perpage@#2\endcsname{%
1797     \ZREF@perpage@step{#2}{#1}%
1798   }%
1799   \expandafter\g@addto@macro\csname @stepcounterhook@#2\endcsname{%
1800     \ifcsname ZREF@perpage@#2\endcsname
1801       \csname ZREF@perpage@#2\endcsname
1802     \fi
1803   }%
1804 }

```

\ZREF@@perpage@step The heart of this module follows.

```

1805 \def\ZREF@@perpage@step#1#2{%

```

First the reference is generated.

```

1806   \global\advance\c@zref@unique\ltx@one
1807   \begingroup

```

```

1808 \expandafter
1809 \zref@labelbylist\expandafter{\thezref@unique}{perpage}%

```

The `\expandafter` commands are necessary, because `\ZREF@temp` is also used inside of `\zref@labelbylist`.

The evaluation of the reference follows. If the reference is not yet known, we use the page counter as approximation.

```

1810 \zref@ifrefundefined\thezref@unique{%
1811   \global\c@zpage=\c@page
1812   \global\let\thezpage\thepage
1813   \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1814     \number\c@abspage
1815   }%
1816 }%

```

The reference is used to set `\thezpage` and counter `zpage`.

```

1817 \global\c@zpage=\zref@extract\thezref@unique{pagevalue}\relax
1818 \xdef\thezpage{\noexpand\zref@extract{\thezref@unique}{page}}%
1819 \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1820   \zref@extractdefault\thezref@unique
1821   {abspage}{\number\c@abspage}%
1822 }%
1823 }%

```

Page changes are detected by a changed absolute page number.

```

1824 \expandafter\ifx\csname ZREF@abspage@#1\endcsname
1825   \csname ZREF@currentabspage@#1\endcsname
1826 \else
1827   \global\csname c@#1\endcsname=#2\relax
1828   \global\expandafter\let
1829     \csname ZREF@currentabspage@#1\endcsname
1830     \csname ZREF@abspage@#1\endcsname
1831 \fi
1832 \endgroup
1833 }

```

(End of definition for `\ZREF@@perpage@step`.)

`\zunmakeperpage` Macro `\zunmakeperpage` cancels the effect of `\zmakeperpage`.

```

1834 \ZREF@ifDefinable\zunmakeperpage\def{%
1835   #1{%
1836     \global\expandafter
1837     \let\csname ZREF@perpage@#1\endcsname\@undefined
1838   }%
1839 }

```

(End of definition for `\zunmakeperpage`. This function is documented on page 18.)

```

1840 \</perpage>

```

6.15 Module `titleref`

```

1841 <*titleref>
1842 \NeedsTeXFormat{LaTeX2e}
1843 \ProvidesPackage{zref-titleref}%
1844   [2026-09-16 v2.38 Module titleref for zref (HO)]%
1845 \RequirePackage{zref-base}[2019/11/29]
1846 \ifx\ZREF@base@ok Y%
1847 \else
1848   \expandafter\endinput
1849 \fi
1850 \RequirePackage{getttitlestring}[2009/12/08]

```

6.15.1 Implementation

```
1851 \RequirePackage{keyval}
```

This module makes section and caption titles available for the reference system. It uses some of the ideas of package `nameref` and `titleref`.

Now we can add the property `title` is added to the main property list.

```
1852 \ZREF@NewPropTitle
1853 \zref@addprop\ZREF@mainlist{title}%
```

The title strings go into the `.aux` file, thus they need some kind of protection. Package `titleref` uses a protected expansion method. The advantage is that this can be used to cleanup the string and to remove `\label`, `\index` and other macros unwanted for referencing. But there is the risk that fragile stuff can break.

Therefore package `nameref` does not expand the string. Thus the entries can safely be written to the `.aux` file. But potentially dangerous macros such as `\label` remain in the string and can cause problems when using the string in references.

`\ifzref@titleref@expand` The switch `\ifzref@titleref@expand` distinguishes between the both methods. Package `nameref`'s behaviour is achieved by setting the switch to false, otherwise `titleref`'s expansion is used. Default is false.

```
1854 \newif\ifzref@titleref@expand
```

(End of definition for \ifzref@titleref@expand.)

`\ZREF@titleref@hook` The hook `\ZREF@titleref@hook` allows to extend the cleanup for the expansion method. Thus unnecessary macros can be removed or dangerous commands removed. The hook is executed before the expansion of `\zref@titleref@current`.

```
1855 \let\ZREF@titleref@hook\ltx@empty
```

(End of definition for \ZREF@titleref@hook.)

`\zref@titleref@cleanup` The hook should not be used directly, instead we provide the macro `\zref@titleref@cleanup` (through a key) to add stuff to the hook and prevents that a previous non-empty content is not discarded accidentally.

```
1856 \ZREF@Robust\def\zref@titleref@cleanup#1{%
1857   \begingroup
1858   \toks@\expandafter{%
1859     \ZREF@titleref@hook
1860     #1%
1861   }%
1862   \expandafter\endgroup
1863   \expandafter\def\expandafter\ZREF@titleref@hook\expandafter{%
1864     \the\toks@
1865   }%
1866 }%
```

(End of definition for \zref@titleref@cleanup.)

`\ifzref@titleref@stripperperiod` Sometimes a title contains a period at the end. Package `nameref` removes this. This behaviour is controlled by the switch `\ifzref@titleref@stripperperiod` and works regardless of the setting of option `expand`. Period stripping is the default.

```
1867 \newif\ifzref@titleref@stripperperiod
1868 \zref@titleref@stripperperiodtrue
```

(End of definition for \ifzref@titleref@stripperperiod.)

`\zref@titleref@setcurrent` Macro `\zref@titleref@setcurrent` sets a new current title stored in `\zref@titleref@current` (and with `\DocumentMetadata` also in `\@currentlabelname`). Some cleanup and expansion is performed that can be controlled by the previous switches.

```
1869 \ZREF@Robust\def\zref@titleref@setcurrent#1{%
1870   \ifzref@titleref@expand
1871     \GetTitleStringExpand{#1}%
```

```

1872 \else
1873   \GetTitleStringNonExpand{#1}%
1874 \fi
1875 \edef\zref@titleref@current{%
1876   \detokenize\expandafter{\GetTitleStringResult}%
1877 }%
1878 \ifzref@titleref@stripperperiod
1879   \edef\zref@titleref@current{%
1880     \expandafter\ZREF@stripperperiod\zref@titleref@current
1881     \ltx@empty.\ltx@empty\@nil
1882   }%
1883 \fi
1884 }%
1885 \IfDocumentMetadataT
1886 {%
1887   \ZREF@Robust\def\zref@titleref@setcurrent#1{%
1888     \ifzref@titleref@expand
1889       \GetTitleStringExpand{#1}%
1890     \else
1891       \GetTitleStringNonExpand{#1}%
1892     \fi
1893     \edef\zref@titleref@current{%
1894       \detokenize\expandafter{\GetTitleStringResult}%
1895     }%
1896     \ifzref@titleref@stripperperiod
1897       \edef\zref@titleref@current{%
1898         \expandafter\ZREF@stripperperiod\zref@titleref@current
1899         \ltx@empty.\ltx@empty\@nil
1900       }%
1901     \fi
1902     \let\@currentlabelname\zref@titleref@current
1903   }%
1904 }
1905 \GetTitleStringDisableCommands{%
1906   \ZREF@titleref@hook
1907 }

```

(End of definition for `\zref@titleref@setcurrent`.)

`\ZREF@stripperperiod` If `\ZREF@stripperperiod` is called, the argument consists of space tokens and tokens with catcode 12 (other), because of ε -TeX's `\detokenize`.

```

1908 \def\ZREF@stripperperiod#1.\ltx@empty#2\@nil{#1}%

```

(End of definition for `\ZREF@stripperperiod`.)

6.15.2 User interface

`\ztitlerefsetup` The behaviour of module `titleref` is controlled by switches and a hook. They can be set by `\ztitlerefsetup` with a key value interface, provided by package `keyval`. Also the current title can be given explicitly by the key `title`.

```

1909 \define@key{ZREF@TR}{expand}[true]{%
1910   \csname zref@titleref@expand#1\endcsname
1911 }%
1912 \define@key{ZREF@TR}{stripperperiod}[true]{%
1913   \csname zref@titleref@stripperperiod#1\endcsname
1914 }%
1915 \define@key{ZREF@TR}{cleanup}{%
1916   \zref@titleref@cleanup{#1}%
1917 }%
1918 \IfDocumentMetadataTF
1919 {
1920   \define@key{ZREF@TR}{title}{%

```

```

1921     \def\zref@titleref@current{#1}%
1922     \let\@currentlabelname\zref@titleref@current
1923   }%
1924 }
1925 {
1926   \define@key{ZREF@TR}{title}{%
1927     \def\zref@titleref@current{#1}%
1928   }%
1929 }
1930 \ZREF@ifDefinable\ztitlerefsetup\def{%
1931   {\kvsetkeys{ZREF@TR}}}%
1932 }%

```

(End of definition for \ztitlerefsetup. This function is documented on page 18.)

\ztitleref The user command \ztitleref references the title. For safety \label is disabled to prevent multiply defined references.

```

1933 \ZREF@ifDefinable\ztitleref\def{%
1934   {\zref@wrapper@babel\ZREF@titleref}%
1935 }%
1936 \def\ZREF@titleref#1{%
1937   \begingroup
1938     \zref@refused{#1}%
1939     \let\label\ltx@gobble
1940     \zref@extract{#1}{title}%
1941   \endgroup
1942 }%

```

(End of definition for \ztitleref. This function is documented on page 18.)

6.15.3 Patches for section and caption commands

The section and caption macros are patched to extract the title data.

Captions of figures and tables.

```

1943 \AtBeginDocument{%
1944   \ZREF@patch{@caption}{%
1945     \long\def\@caption#1[#2]{%
1946       \zref@titleref@setcurrent{#2}%
1947       \ZREF@org@@caption{#1}[#2]}%
1948   }%
1949 }%

```

Section commands without star. The title version for the table of contents is used because it is usually shorter and more robust.

```

1950 \ZREF@patch{@part}{%
1951   \def\@part[#1]{%
1952     \zref@titleref@setcurrent{#1}%
1953     \ZREF@org@@part[#1]}%
1954   }%
1955 }%
1956 \ZREF@patch{@chapter}{%
1957   \def\@chapter[#1]{%
1958     \zref@titleref@setcurrent{#1}%
1959     \ZREF@org@@chapter[#1]}%
1960   }%
1961 }%
1962 \ZREF@patch{@sect}{%
1963   \def\@sect#1#2#3#4#5#6[#7]{%
1964     \zref@titleref@setcurrent{#7}%
1965     \ZREF@org@@sect{#1}{#2}{#3}{#4}{#5}{#6}[#7]}%
1966   }%
1967 }%

```

The star versions of the section commands.

```

1968 \ZREF@patch{@spart}{%
1969   \def\@spart#1{%
1970     \zref@titleref@setcurrent{#1}%
1971     \ZREF@org@@spart{#1}%
1972   }%
1973 }%
1974 \ZREF@patch{@schapter}{%
1975   \def\@schapter#1{%
1976     \zref@titleref@setcurrent{#1}%
1977     \ZREF@org@@schapter{#1}%
1978   }%
1979 }%
1980 \ZREF@patch{@ssect}{%
1981   \def\@ssect#1#2#3#4#5{%
1982     \zref@titleref@setcurrent{#5}%
1983     \ZREF@org@@ssect{#1}{#2}{#3}{#4}{#5}%
1984   }%
1985 }%

```

6.15.4 Environment description

```

1986 \ZREF@patch{descriptionlabel}{%
1987   \def\descriptionlabel#1{%
1988     \zref@titleref@setcurrent{#1}%
1989     \ZREF@org@descriptionlabel{#1}%
1990   }%
1991 }%

```

6.15.5 Class memoir

```

1992 \@ifclassloaded{memoir}{%
1993   \ltx@ifundefined{ifheadnameref}{-}{%
1994     \def\@chapter[#1]#2{%
1995       \ltx@ifundefined{ch@pt@c}{%
1996         \zref@titleref@setcurrent{#1}%
1997       }{%
1998         \ifx\ch@pt@c\ltx@empty
1999           \zref@titleref@setcurrent{#2}%
2000         \else
2001           \def\NR@temp{#1}%
2002           \ifx\NR@temp\ltx@empty
2003             \expandafter\zref@titleref@setcurrent
2004             \expandafter{\ch@pt@c}%
2005           \else
2006             \ifheadnameref
2007               \zref@titleref@setcurrent{#1}%
2008             \else
2009               \expandafter\zref@titleref@setcurrent
2010               \expandafter{\ch@pt@c}%
2011             \fi
2012           \fi
2013         \fi
2014       }%
2015       \ZREF@org@@chapter[{#1}]{#2}%
2016     }%
2017   \ZREF@patch{M@ssect}{%
2018     \def\M@ssect#1#2#3#4#5#6[#7][#8]{%
2019       \ifheadnameref
2020         \zref@titleref@setcurrent{#8}%
2021       \else
2022         \zref@titleref@setcurrent{#7}%

```

```

2023         \fi
2024         \ZREF@org@M@sect{#1}{#2}{#3}{#4}{#5}{#6} [{#7}] [{#8}]%
2025     }%
2026 }%
2027 }%
2028 }{}%

```

6.15.6 Class beamer

```

2029 \@ifclassloaded{beamer}{%
2030     \ZREF@patch{beamer@section}{%
2031         \long\def\beamer@section[#1]{%
2032             \zref@titleref@setcurrent{#1}%
2033             \ZREF@org@beamer@section[{#1}]%
2034         }%
2035     }%
2036     \ZREF@patch{beamer@subsection}{%
2037         \long\def\beamer@subsection[#1]{%
2038             \zref@titleref@setcurrent{#1}%
2039             \ZREF@org@beamer@subsection[{#1}]%
2040         }%
2041     }%
2042     \ZREF@patch{beamer@subsubsection}{%
2043         \long\def\beamer@subsubsection[#1]{%
2044             \zref@titleref@setcurrent{#1}%
2045             \ZREF@org@beamer@subsubsection[{#1}]%
2046         }%
2047     }%
2048 }{}%

```

6.15.7 Package titlesec

```

2049 \@ifpackageloaded{titlesec}{%
2050     \ZREF@patch{ttl@sect@i}{%
2051         \def\ttl@sect@i#1#2[#3]#4{%
2052             \zref@titleref@setcurrent{#4}%
2053             \ZREF@org@ttl@sect@i{#1}{#2} [{#3}] {#4}%
2054         }%
2055     }%
2056     \ZREF@patch{ttl@straight@i}{%
2057         \def\ttl@straight@i#1[#2]#3{%
2058             \def\ZREF@temp{#2}%
2059             \ifx\ZREF@temp\ltx@empty
2060                 \zref@titleref@setcurrent{#3}%
2061             \else
2062                 \zref@titleref@setcurrent{#2}%
2063             \fi
2064             \ZREF@org@ttl@straight@i{#1} [{#2}] {#3}%
2065         }%
2066     }%
2067 }{}%

```

6.15.8 Package longtable

Package longtable: some support for its `\caption`. However `\label` inside the caption is not supported.

```

2068 \@ifpackageloaded{longtable}{%
2069     \ZREF@patch{LT@c@ption}{%
2070         \def\LT@c@ption#1[#2]#3{%
2071             \ZREF@org@LT@c@ption{#1} [{#2}] {#3}%
2072             \zref@titleref@setcurrent{#2}%
2073         }%

```

```

2074     }%
2075 }{}%

```

6.15.9 Package listings

Package listings: support for its caption.

```

2076 \ifpackageloaded{listings}{%
2077   \ZREF@patch{lst@MakeCaption}{%
2078     \def\lst@MakeCaption{%
2079       \ifx\lst@label\ltx@empty
2080       \else
2081         \expandafter\zref@titleref@setcurrent\expandafter{%
2082           \lst@caption
2083         }%
2084       \fi
2085       \ZREF@org@lst@MakeCaption
2086     }%
2087   }%
2088 }{}%

```

6.15.10 Theorems

```

2089 \ZREF@patch{@opargbegintheorem}{%
2090   \def\@opargbegintheorem#1#2#3{%
2091     \zref@titleref@setcurrent{#3}%
2092     \ZREF@org@@opargbegintheorem{#1}{#2}{#3}%
2093   }%
2094 }%
2095 \ifpackageloaded{amsthm}{%
2096   \begingroup
2097     \edef\x{macro:\string#1\string#2[\string#3]}%
2098     \@onelevel@sanitize\x
2099     \def\y#1->#2\@nil{#1}%
2100     \edef\z{\expandafter\y\meaning\@begintheorem->\@nil}%
2101     \@onelevel@sanitize\z
2102     \expandafter\endgroup
2103     \ifx\x\z
2104       \ZREF@patch{@begintheorem}{%
2105         \def\@begintheorem#1#2[#3]{%
2106           \zref@titleref@setcurrent{#3}%
2107           \ZREF@org@@begintheorem{#1}{#2}[\string#3]}%
2108         }%
2109       }%
2110     \fi
2111   }{}%
2112 }
2113 </titleref>

```

6.16 Module xr

```

2114 <*xr>
2115 \NeedsTeXFormat{LaTeX2e}
2116 \ProvidesPackage{zref-xr}%
2117   [2026-09-16 v2.38 Module xr for zref (H0)]%
2118 \RequirePackage{zref-base}[2019/11/29]
2119 \ifx\ZREF@base@ok Y%
2120 \else
2121   \expandafter\endinput
2122 \fi

```

```

2123 \RequirePackage{keyval}
2124 \RequirePackage{kvoptions}[2010/02/22]

```

We declare property `url`, because this is added, if a reference is imported and has not already set this field. Or if `hyperref` is used, then this property can be asked.

```

2125 \zref@newprop{url}{}%
2126 \zref@newprop{urluse}{}%

2127 \zref@newprop{externaldocument}{}%

```

Most code, especially the handling of the `.aux` files are taken from David Carlisle's `xr` package. Therefore I drop the documentation for these macros here.

\zref@xr@ext If the URL is not specified, then assume processed file with a guessed extension. Use the setting of `hyperref` if available.

```

2128 \providecommand*\zref@xr@ext{%
2129   \ltx@ifundefined{XR@ext}{pdf}{\XR@ext}%
2130 }%

```

(End of definition for \zref@xr@ext. This function is documented on page 21.)

\ifZREF@xr@zreflabel The use of the star form of `\zexternaldocument` is remembered in the switch `\ifZREF@xr@zreflabel`.

```

2131 \newif\ifZREF@xr@zreflabel

```

(End of definition for \ifZREF@xr@zreflabel.)

```

2132 \SetupKeyvalOptions{%
2133   family=ZREF@XR,%
2134   prefix=ZREF@xr%
2135 }
2136 \DeclareBoolOption[true]{tozreflabel}
2137 \DeclareBoolOption[false]{toltxlabel}
2138 \DeclareBoolOption{verbose}
2139 \define@key{ZREF@XR}{ext}{%
2140   \def\zref@xr@{#1}%
2141 }
2142 \DeclareBoolOption[false]{urluse}

```

\zxrsetup

```

2143 \newcommand*\zxrsetup{%
2144   \kvsetkeys{ZREF@XR}%
2145 }%

```

(End of definition for \zxrsetup. This function is documented on page 21.)

\ZREF@xr@URL

```

2146 \newcount\ZREF@xr@URL
2147 \ZREF@xr@URL=\ltx@zero

```

(End of definition for \ZREF@xr@URL.)

\ZREF@xr@AddURL

```

2148 \def\ZREF@xr@AddURL#1{%
2149   \begingroup
2150     \def\ZREF@temp{#1}%
2151     \count@=\ltx@one
2152     \ZREF@xr@@AddUrl
2153   \endgroup
2154 }

```

(End of definition for \ZREF@xr@AddURL.)

`\ZREF@xr@@AddUrl`

```

2155 \def\ZREF@xr@@AddUrl{%
2156   \ifnum\count@>\ZREF@xr@URL
2157     \global\advance\ZREF@xr@URL by\ltx@one
2158     \xdef\ZREF@xr@theURL{\romannumeral\ZREF@xr@URL}%
2159     \global\expandafter\let
2160       \csname Z@U@\ZREF@xr@theURL\endcsname\ZREF@temp
2161     \@PackageInfo{zref-xr}{%
2162       \ltx@backslashchar Z@U@\ZREF@xr@theURL:\MessageBreak
2163       \ZREF@temp\MessageBreak
2164     }%
2165   \else
2166     \expandafter
2167       \ifx\csname Z@U@\romannumeral\count@\endcsname\ZREF@temp
2168       \xdef\ZREF@xr@theURL{\romannumeral\count@}%
2169     \else
2170       \expandafter\expandafter\expandafter\ZREF@xr@@AddUrl
2171     \fi
2172   \fi
2173 }

```

(End of definition for `\ZREF@xr@@AddUrl`.)

`\zexternaldocument` In its star form it looks for `\newlabel`, otherwise for `\zref@newlabel`. Later we will read `.aux` files that expects `@` to have catcode 11 (letter).

```

2174 \ZREF@ifDefinable\zexternaldocument\def{%
2175   {%
2176     \ZREF@NewPropAnchor
2177     \ZREF@NewPropTitle
2178     \begingroup
2179       \csname @safe@actives@true\endcsname
2180       \makeatletter
2181       \@ifstar{%
2182         \ZREF@xr@zreflabelfalse
2183         \@testopt\ZREF@xr@externaldocument{}%
2184       }{%
2185         \ZREF@xr@zreflabeltrue
2186         \@testopt\ZREF@xr@externaldocument{}%
2187       }%
2188     }%
2189   }%

```

(End of definition for `\zexternaldocument`. This function is documented on page 21.)

If the `\include` feature was used, there can be several `.aux` files. These files are read one after another, especially they are not recursively read in order to save read registers. Thus it can happen that the read order of the `newlabel` commands differs from L^AT_EX's order using `\input`.

`\ZREF@xr@externaldocument` It reads the remaining arguments. `\newcommand` comes in handy for the optional argument.

```

2190 \def\ZREF@xr@externaldocument[#1]#2{%
2191   \def\ZREF@xr@prefix{#1}%
2192   \let\ZREF@xr@filelist\ltx@empty
2193   \edef\ZREF@xr@externalfile{#2}%
2194   \edef\ZREF@xr@file{\ZREF@xr@externalfile.aux}%
2195   \filename@parse{#2}%
2196   \@testopt\ZREF@xr@graburl{#2.\zref@xr@ext}%
2197 }%
2198 \def\ZREF@xr@graburl[#1]{%
2199   \edef\ZREF@xr@url{#1}%
2200   \ifZREF@xr@urluse
2201     \expandafter\ZREF@xr@AddURL\expandafter{\ZREF@xr@url}%

```

```

2202     \expandafter\def\expandafter\ZREF@xr@url
2203     \expandafter{\csname Z@U@\ZREF@xr@theURL\endcsname}%
2204     \fi
2205     \ZREF@xr@checkfile
2206   \endgroup
2207 }%

```

(End of definition for \ZREF@xr@externaldocument.)

\ZREF@xr@processfile We follow xr here, \IfFileExists offers a nicer test, but we have to open the file anyway.

```

2208 \def\ZREF@xr@checkfile{%
2209   \openin\@inputcheck\ZREF@xr@file\relax
2210   \ifeof\@inputcheck
2211     \PackageWarning{zref-xr}{%
2212       File '\ZREF@xr@file' not found or empty,\MessageBreak
2213       labels not imported%
2214     }%
2215   \else
2216     \PackageInfo{zref-xr}{%
2217       Label \ifZREF@xr@zreflabel (zref) \fi
2218       import from '\ZREF@xr@file'%
2219     }%
2220     \def\ZREF@xr@found{0}%
2221     \def\ZREF@xr@ignored@empty{0}%
2222     \def\ZREF@xr@ignored@zref{0}%
2223     \def\ZREF@xr@ignored@ltx{0}%
2224     \ZREF@xr@processfile
2225     \closein\@inputcheck
2226     \begingroup
2227       \let\on@line\ltx@empty
2228       \PackageInfo{zref-xr}{%
2229         Statistics for '\ZREF@xr@file':\MessageBreak
2230         \ZREF@xr@found\space
2231         \ifZREF@xr@zreflabel zref\else LaTeX\fi\space
2232         label(s) found%
2233         \ifnum\ZREF@xr@ignored@empty>0 %
2234           ,\MessageBreak
2235           \ZREF@xr@ignored@empty\space empty label(s) ignored%
2236         \fi
2237         \ifnum\ZREF@xr@ignored@zref>0 %
2238           ,\MessageBreak
2239           \ZREF@xr@ignored@zref\space
2240           duplicated zref label(s) ignored%
2241         \fi
2242         \ifnum\ZREF@xr@ignored@ltx>0 %
2243           ,\MessageBreak
2244           \ZREF@xr@ignored@ltx\space
2245           duplicated latex label(s) ignored%
2246         \fi
2247       }%
2248     \endgroup
2249   \fi
2250   \ifx\ZREF@xr@filelist\ltx@empty
2251   \else
2252     \edef\ZREF@xr@file{%
2253       \expandafter\ltx@car\ZREF@xr@filelist\@nil
2254     }%
2255     \edef\ZREF@xr@filelist{%
2256       \expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil
2257     }%
2258     \expandafter\ZREF@xr@checkfile
2259   \fi
2260 }%

```

(End of definition for \ZREF@xr@processfile.)

\ZREF@xr@processfile

```

2261 \def\ZREF@xr@processfile{%
2262   \read\@inputcheck to\ZREF@xr@line
2263   \expandafter\ZREF@xr@processline\ZREF@xr@line..\ZREF@nil
2264   \ifeof\@inputcheck
2265   \else
2266     \expandafter\ZREF@xr@processfile
2267   \fi
2268 }%
```

(End of definition for \ZREF@xr@processfile.)

\ZREF@xr@processline The most work must be done for analyzing the arguments of \newlabel.

```

2269 \long\def\ZREF@xr@processline#1#2#3\ZREF@nil{%
2270   \def\x{#1}%
2271   \toks@{#2}%
2272   \ifZREF@xr@zreflabel
2273     \ifx\x\ZREF@xr@zref@newlabel
2274       \expandafter
2275         \ZREF@xr@process@zreflabel\ZREF@xr@line...\ZREF@nil
2276     \fi
2277   \else
2278     \ifx\x\ZREF@xr@newlabel
2279       \expandafter
2280         \ZREF@xr@process@label\ZREF@xr@line...[]\ZREF@nil
2281     \fi
2282   \fi
2283   \ifx\x\ZREF@xr@@input
2284     \edef\ZREF@xr@filelist{%
2285       \etex@unexpanded\expandafter{\ZREF@xr@filelist}%
2286       {\filename@area\the\toks@}%
2287     }%
2288   \fi
2289 }%
2290 \def\ZREF@xr@process@zreflabel\zref@newlabel#1#2#3\ZREF@nil{%
2291   \edef\ZREF@xr@refname{Z@R@ZREF@xr@prefix#1}%
2292   \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2293   \def\x{#2}%
2294   \edef\ZREF@xr@tempname{$temp$}%
2295   \edef\ZREF@xr@temprefname{Z@R@ZREF@xr@tempname}%
2296   \let\ZREF@xr@list\x
2297   \ifx\ZREF@xr@list\ltx@empty
2298     \PackageWarningNoLine{zref-xr}{%
2299       Label `#1' without properties ignored\MessageBreak
2300       in file ``\ZREF@xr@file'%
2301     }%
2302     \edef\ZREF@xr@ignored@empty{%
2303       \the\numexpr\ZREF@xr@ignored@empty+1\relax
2304     }%
2305   \else
2306     \expandafter\ZREF@xr@checklist\x\ZREF@nil
2307     \expandafter\let\csname\ZREF@xr@temprefname\endcsname\x
2308     \expandafter\ltx@LocalAppendToMacro
2309     \csname\ZREF@xr@temprefname\expandafter\endcsname
2310     \expandafter{%
2311       \expandafter\externaldocument\expandafter{%
2312         \ZREF@xr@externalfile
2313       }%
2314     }%
2315     \ZREF@xr@urlcheck\ZREF@xr@tempname
2316     \ifZREF@xr@tozreflabel
```

```

2317 \ifundefined{\ZREF@xr@refname}{%
2318 \ifZREF@xr@verbose
2319 \PackageInfo{zref-xr}{%
2320 Import to zref label `~\ZREF@xr@tempname#1'%
2321 }%
2322 \fi
2323 \global\expandafter
2324 \let\csname\ZREF@xr@refname\expandafter\endcsname
2325 \csname\ZREF@xr@temprefname\endcsname
2326 }{%
2327 \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2328 }%
2329 \fi
2330 \ifZREF@xr@toltxlabel
2331 \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2332 \fi
2333 \fi
2334 }%
2335 \def\ZREF@xr@process@label\newlabel#1#2#3[#4]#5\ZREF@nil{%
2336 \def\ZREF@xr@refname{Z@R@~\ZREF@xr@prefix#1}%
2337 \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2338 \def\x{#2}%
2339 \edef\ZREF@xr@tempname{$temp$}%
2340 \edef\ZREF@xr@temprefname{Z@R@~\ZREF@xr@tempname}%
2341 \expandafter\ZREF@xr@scanparams
2342 \csname\ZREF@xr@temprefname\expandafter\endcsname
2343 \x{}{}{}{}{}~\ZREF@nil
2344 \ifx\\#4\\%
2345 \else
2346 % ntheorem knows an optional argument at the end of \newlabel
2347 \ZREF@NewPropTheotype
2348 \expandafter\ltx@LocalAppendToMacro
2349 \csname\ZREF@xr@temprefname\endcsname{\theotype{#4}}%
2350 \fi
2351 \expandafter\ltx@LocalAppendToMacro
2352 \csname\ZREF@xr@temprefname\expandafter\endcsname\expandafter{%
2353 \expandafter\externaldocument\expandafter{%
2354 \ZREF@xr@externalfile
2355 }%
2356 }%
2357 \ZREF@xr@urlcheck\ZREF@xr@tempname
2358 \ifZREF@xr@tozreflabel
2359 \ifundefined{\ZREF@xr@refname}{%
2360 \ifZREF@xr@verbose
2361 \PackageInfo{zref-xr}{%
2362 Import to zref label `~\ZREF@xr@prefix#1'%
2363 }%
2364 \fi
2365 \global\expandafter
2366 \let\csname\ZREF@xr@refname\expandafter\endcsname
2367 \csname\ZREF@xr@temprefname\endcsname
2368 }{%
2369 \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2370 }%
2371 \fi
2372 \ifZREF@xr@toltxlabel
2373 \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2374 \fi
2375 }
2376 \def\ZREF@xr@zref@newlabel{\zref@newlabel}%
2377 \def\ZREF@xr@newlabel{\newlabel}%
2378 \def\ZREF@xr@@input{\@input}%

```

```

2379 \def\ZREF@xr@relax{\relax}%
(End of definition for \ZREF@xr@processline.)

```

\ZREF@xr@tolabel

```

2380 \def\ZREF@xr@tolabel#1#2{%
2381   \ifZREF@xr@verbose
2382     \PackageInfo{zref-xr}{%
2383       Import to LaTeX label `#2'%
2384     }%
2385   \fi
2386   \zref@wrapper@unexpanded{%
2387     \expandafter\xdef\csname r@#2\endcsname{%
2388       {%
2389         \ltx@ifundefined{M@TitleReference}{%
2390           \ltx@ifundefined{TR@TitleReference}{%
2391             \zref@extractdefault{#1}{default}{}}%
2392           {%
2393             \noexpand\TR@TitleReference
2394             {\zref@extractdefault{#1}{default}{}}}%
2395             {\zref@extractdefault{#1}{title}{}}}%
2396           }%
2397         {%
2398           \noexpand\M@TitleReference
2399           {\zref@extractdefault{#1}{default}{}}}%
2400           {\zref@extractdefault{#1}{title}{}}}%
2401         }%
2402       }%
2403       {\zref@extractdefault{#1}{page}{}}}%
2404     \ltx@ifpackageloaded{nameref}{%
2405       {\zref@extractdefault{#1}{title}{}}}%
2406       {\zref@extractdefault{#1}{anchor}{}}}%
2407     \zref@ifrefcontainsprop{#1}{urluse}{%
2408       {\zref@extractdefault{#1}{urluse}{}}}%
2409     }{%
2410       {\zref@extractdefault{#1}{url}{}}}%
2411     }%
2412   }{}%
2413 }%
2414 }%
2415 }

```

(End of definition for \ZREF@xr@tolabel.)

\ZREF@xr@zref@ignorewarning

```

2416 \def\ZREF@xr@zref@ignorewarning#1{%
2417   \PackageWarningNoLine{zref-xr}{%
2418     Zref label `#1' is already in use\MessageBreak
2419     in file ``\ZREF@xr@file'%
2420   }%
2421   \edef\ZREF@xr@ignored@zref{%
2422     \the\numexpr\ZREF@xr@ignored@zref+1%
2423   }%
2424 }%

```

(End of definition for \ZREF@xr@zref@ignorewarning.)

\ZREF@xr@ltx@ignorewarning

```

2425 \def\ZREF@xr@ltx@ignorewarning#1{%
2426   \PackageWarningNoLine{zref-xr}{%
2427     LaTeX label `#1' is already in use\MessageBreak
2428     in file ``\ZREF@xr@file'%
2429   }%
2430   \edef\ZREF@xr@ignored@ltx{%

```

```

2431 \the\numexpr\ZREF@xr@ignored@ltx+1%
2432 }%
2433 }%

```

(End of definition for \ZREF@xr@ltx@ignorewarning.)

\ZREF@xr@checklist

```

2434 \def\ZREF@xr@checklist#1#2#3\ZREF@nil{%
2435 \ifx\@undefined#1\relax
2436 \expandafter\ZREF@xr@checkkey\string#1\@nil
2437 \fi
2438 \ifx\#3\%
2439 \else
2440 \ltx@ReturnAfterFi{%
2441 \ZREF@xr@checklist#3\ZREF@nil
2442 }%
2443 \fi
2444 }%
2445 \def\ZREF@xr@checkkey#1#2\@nil{%
2446 \zref@ifpropundefined{#2}{%
2447 \zref@newprop{#2}{}%
2448 }{}%
2449 }%

```

(End of definition for \ZREF@xr@checklist.)

\ZREF@xr@scanparams

```

2450 \def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{%
2451 \let#1\ltx@empty
2452 \ZREF@foundfalse
2453 \ZREF@xr@scantitleref#1#2\TR@TitleReference{}{}\ZREF@nil
2454 \ifZREF@found
2455 \else
2456 \ltx@LocalAppendToMacro#1{\default{#2}}%
2457 \fi
2458 % page
2459 \ltx@LocalAppendToMacro#1{\page{#3}}%
2460 % nameref title
2461 \ifZREF@found
2462 \else
2463 \ifx\#4\%
2464 \else
2465 \def\ZREF@xr@temp{#4}%
2466 \ifx\ZREF@xr@temp\ZREF@xr@relax
2467 \else
2468 \ltx@LocalAppendToMacro#1{\title{#4}}%
2469 \fi
2470 \fi
2471 \fi
2472 % anchor
2473 \ifx\#5\%
2474 \else
2475 \ltx@LocalAppendToMacro#1{\anchor{#5}}%
2476 \fi
2477 \ifx\#6\%
2478 \else
2479 \ifZREF@xr@urluse
2480 \ZREF@xr@AddURL{#6}%
2481 \expandafter\ltx@LocalAppendToMacro\expandafter#1%
2482 \expandafter{%
2483 \expandafter\urluse\expandafter{%
2484 \csname Z@U@\ZREF@xr@theURL\endcsname
2485 }%

```

```

2486     }%
2487     \else
2488         \ltx@LocalAppendToMacro#1{\url{#6}}%
2489     \fi
2490     \fi
2491 }%

```

(End of definition for \ZREF@xr@scanparams.)

\ZREF@xr@scantitleref

```

2492 \def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{%
2493     \ifx\#5\%
2494     \else
2495         \ltx@LocalAppendToMacro#1{%
2496             \default{#3}%
2497             \title{#4}%
2498         }%
2499         \ZREF@foundtrue
2500     \fi
2501 }%

```

(End of definition for \ZREF@xr@scantitleref.)

\ZREF@xr@urlcheck

```

2502 \def\ZREF@xr@urlcheck#1{%
2503     \zref@ifrefcontainsprop{#1}{anchor}{%
2504         \zref@ifrefcontainsprop{#1}{url}{%
2505             }{%
2506                 \expandafter
2507                 \ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
2508                 \expandafter{%
2509                     \csname url\ifZREF@xr@urluse use\fi
2510                     \expandafter\endcsname\expandafter{\ZREF@xr@url}%
2511                 }%
2512             }%
2513         }{%
2514             }%
2515         }%

```

(End of definition for \ZREF@xr@urlcheck.)

```

2516 \</xr>

```

6.17 Module hyperref

UNFINISHED :-(

```

2517 \<hyperref>
2518 \NeedsTeXFormat{LaTeX2e}
2519 \ProvidesPackage{zref-hyperref}%
2520     [2026-09-16 v2.38 Module hyperref for zref (HO)]%
2521 \RequirePackage{zref-base}[2019/11/29]
2522 \ifx\ZREF@base@ok Y%
2523 \else
2524     \expandafter\endinput
2525 \fi

2526 \ZREF@NewPropAnchor
2527 \zref@addprop\ZREF@mainlist{anchor}%

2528 \</hyperref>

```

6.18 Module savepos

Module savepos provides an interface for pdfTeX's \pdfsavepos and similar commands from the other engines, see the manual for pdfTeX.

6.18.1 Identification

```

2529 <*savepos>
2530 \NeedsTeXFormat{LaTeX2e}
2531 \ProvidesPackage{zref-savepos}%
2532 [2026-09-16 v2.38 Module savepos for zref (HO)]%
2533 \RequirePackage{zref-base}[2019/11/29]
2534 \ifx\ZREF@base@ok Y%
2535 \else
2536 \expandafter\endinput
2537 \fi

```

6.18.2 Availability

We now assume that it is available always.

6.18.3 Setup

```

2538 \zref@newlist{savepos}
2539 \ExplSyntaxOn
2540 \zref@newprop*{posx}[0]{\int_use:N \tex_lastxpos:D}
2541 \zref@newprop*{posy}[0]{\int_use:N \tex_lastypos:D}
2542 \zref@addprops{savepos}{posx,posy}

```

6.18.4 User macros

\zref@savepos

```

2543 \def\zref@savepos{%
2544 \if@filesw
2545 \tex_savepos:D
2546 \fi}
2547 \ExplSyntaxOff

```

(End of definition for \zref@savepos. This function is documented on page 19.)

\ZREF@zsavepos

```

2548 \def\ZREF@zsavepos#1#2#3{%
2549 \@bsphack
2550 \if@filesw
2551 \zref@savepos
2552 #1{#3}{#2}%
2553 \zref@savepos
2554 \fi
2555 \@esphack
2556 }

```

(End of definition for \ZREF@zsavepos.)

\zsavapos The current location is stored in a reference with the given name.

```

2557 \ZREF@IfDefinable\zsavapos\def{%
2558 {%
2559 \ZREF@zsavepos\zref@labelbylist{savepos}%
2560 }%
2561 }

```

(End of definition for \zsavapos. This function is documented on page 19.)

\zsavaposx

```

2562 \ZREF@IfDefinable\zsavaposx\def{%
2563 {%
2564 \ZREF@zsavepos\zref@labelbyprops{posx}%
2565 }%
2566 }

```

(End of definition for `\zsaveposx`. This function is documented on page 19.)

`\zsaveposy`

```
2567 \ZREF@ifDefinable\zsaveposy\def{%
2568   {%
2569     \ZREF@zsavepos\zref@labelbyprops{posy}%
2570   }%
2571 }
```

(End of definition for `\zsaveposy`. This function is documented on page 19.)

`\zposx`

`\zposy`

The horizontal and vertical position are available by `\zposx` and `\zposy`. Do not rely on absolute positions. They differ in DVI and PDF mode of pdfTeX. Use differences instead. The unit of the position numbers is sp.

```
2572 \newcommand*{\zposx}[1]{%
2573   \zref@extract{#1}{posx}%
2574 }%
2575 \newcommand*{\zposy}[1]{%
2576   \zref@extract{#1}{posy}%
2577 }%
```

(End of definition for `\zposx` and `\zposy`. These functions are documented on page 19.)

Typically horizontal and vertical positions are used inside calculations. Therefore the extracting macros should be expandable and babel’s patch is not applicable.

Also it is in the responsibility of the user to mark used positions by `\zrefused` in order to notify L^AT_EX about undefined references.

`\ZREF@savepos@ok`

```
2578 \let\ZREF@savepos@ok=Y
```

(End of definition for `\ZREF@savepos@ok`.)

```
2579 </savepos>
```

6.19 Module `abspos`

6.19.1 Identification

```
2580 <*abspos>
2581 \NeedsTeXFormat{LaTeX2e}
2582 \ProvidesPackage{zref-abspos}%
2583   [2026-09-16 v2.38 Module abspos for zref (H0)]%
2584 \RequirePackage{zref-base}[2019/11/29]
2585 \ifx\ZREF@base@ok Y%
2586   \else
2587     \expandafter\endinput
2588   \fi
2589 \RequirePackage{zref-savepos}[2019/11/29]
2590 \ifx\ZREF@savepos@ok Y%
2591   \else
2592     \expandafter\endinput
2593   \fi
2594 \RequirePackage{zref-pagelayout}[2019/11/29]
2595 \zref@addprop{savepos}{abspage}
2596 \zref@addprop{savepos}{pagevalue}
```

`\zref@absposx`

```
2597 \newcommand*{\zref@absposx}[3]{%
2598   \number
2599     \expandafter\zref@absposnumx\expandafter{%
2600       \number\zref@extractdefault{#1}{abspage}{0}%
2601     }{#2}{#3}%
2602   \ltx@space
2603 }
```

(End of definition for \zref@absposx. This function is documented on page 20.)

\zref@absposy

```
2604 \newcommand*\zref@absposy}[3]{%
2605   \number
2606   \expandafter\zref@absposnumy\expandafter{%
2607     \number\zref@extractdefault{#1}{abspage}{0}%
2608   }{#2}{#3}%
2609   \ltx@space
2610 }
```

(End of definition for \zref@absposy. This function is documented on page 20.)

\zref@absposnumx

```
2611 \newcommand*\zref@absposnumx}[3]{%
2612   \number
2613   \numexpr\ZREF@absposnum{thepage#1}{#2}{x}{#3}\relax
2614 }
```

(End of definition for \zref@absposnumx. This function is documented on page 20.)

\zref@absposnumy

```
2615 \newcommand*\zref@absposnumy}[3]{%
2616   \number
2617   \numexpr\ZREF@absposnum{thepage#1}{#2}{y}{#3}\relax
2618 }
```

(End of definition for \zref@absposnumy. This function is documented on page 20.)

\ZREF@absposnum

```
2619 \def\ZREF@absposnum#1#2#3#4{%
2620   \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
2621     0%
2622   }{%
2623     \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
2624   }%
2625 }
```

(End of definition for \ZREF@absposnum.)

\zref@def@absposx

```
2626 \ZREF@Robust\def\zref@def@absposx#1{%
2627   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposx}%
2628 }
```

(End of definition for \zref@def@absposx. This function is documented on page 20.)

\zref@def@absposy

```
2629 \ZREF@Robust\def\zref@def@absposy#1{%
2630   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
2631 }
```

(End of definition for \zref@def@absposy. This function is documented on page 20.)

\zref@def@absposnumx

```
2632 \ZREF@Robust\def\zref@def@absposnumx#1{%
2633   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumx}%
2634 }
```

(End of definition for \zref@def@absposnumx. This function is documented on page 20.)

\zref@def@absposnumy

```
2635 \ZREF@Robust\def\zref@def@absposnumy#1{%
2636   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumy}%
2637 }
```

(End of definition for \zref@def@absposnumy. This function is documented on page 20.)

\ZREF@def@abspos

```
2638 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2639   \edef#1{#2{#3}{#4}{#5}}%
2640 }
```

(End of definition for \ZREF@def@abspos.)

\zref@absposused

```
2641 \ZREF@Robust\def\zref@absposused{%
2642   \zref@wrapper@babel\ZREF@abspos@used
2643 }
```

(End of definition for \zref@absposused. This function is documented on page 20.)

\ZREF@abspos@used

```
2644 \def\ZREF@abspos@used#1{%
2645   \zref@refused{#1}%
2646   \zref@ifrefundefined{#1}{%
2647     }{%
2648     \begingroup
2649     \edef\ZREF@temp{%
2650       \zref@extractdefault{#1}{abspage}{0}%
2651     }%
2652     \ifnum\ZREF@temp>\ltx@zero
2653       \zref@refused{thepage\ZREF@temp}%
2654     \else
2655       \@PackageError{zref-abspos}{%
2656         \string\zref@pos@label@used\ltx@space
2657         needs property `abspage'\MessageBreak
2658         in label `#1'%
2659       }\@ehc
2660     \fi
2661   \endgroup
2662 }%
2663 }
```

(End of definition for \ZREF@abspos@used.)

\zref@absposnumused

```
2664 \newcommand*{\zref@absposnumused}[1]{%
2665   \ifnum#1>\ltx@zero
2666     \zref@refused{thepage\number#1}%
2667   \else
2668     \@PackageError{zref-abspos}{%
2669       Invalid absolute page number (#1)\MessageBreak
2670       for \string\zref@pos@num@used.\MessageBreak
2671       A positive integer is expected%
2672     }\@ehc
2673   \fi
2674 }
```

(End of definition for \zref@absposnumused. This function is documented on page 20.)

\zref@ifabsposundefined

```
2675 \def\zref@ifabsposundefined#1{%
2676   \zref@ifrefundefined{#1}\ltx@firsttwo{%
2677     \expandafter\zref@ifabsposnumundefined\expandafter{%
2678       \number\zref@extractdefault{#1}{abspage}{0}%
2679     }%
2680   }%
2681 }
```

(End of definition for \zref@ifabsposundefined. This function is documented on page 20.)

\zref@ifabsposnumundefined

```

2682 \def\zref@ifabsposnumundefined#1{%
2683   \ifnum\ZREF@number{#1}>\ltx@zero
2684     \zref@ifrefundefined{thepage#1}%
2685     \ltx@firstoftwo\ltx@secondoftwo
2686   \else
2687     \expandafter\ltx@firstoftwo
2688   \fi
2689 }

```

(End of definition for \zref@ifabsposnumundefined. This function is documented on page 20.)

6.19.2 Media

Ensure that \stockwidth and \stockheight are defined

```

2690 \@ifundefined{stockwidth}{\newdimen\stockwidth}{}
2691 \@ifundefined{stockheight}{\newdimen\stockheight}{}

```

\ZREF@abspos@media@width

```

2692 \edef\ZREF@abspos@media@width{%
2693   \ltx@ifundefined{pdfpagewidth}{%
2694     \ltx@ifundefined{mediawidth}{%
2695       \ifdim\stockwidth > \z@
2696         stockwidth%
2697       \else
2698         paperwidth%
2699       \fi
2700     }{%
2701       mediawidth%
2702     }%
2703   }{%
2704     pdfpagewidth%
2705   }%
2706 }
2707 \ifluatex
2708 \def\ZREF@abspos@media@width{pdfpagewidth}%
2709 \fi

```

(End of definition for \ZREF@abspos@media@width.)

\ZREF@abspos@media@height

```

2710 \edef\ZREF@abspos@media@height{%
2711   \ltx@ifundefined{pdfpageheight}{%
2712     \ltx@ifundefined{mediaheight}{%
2713       \ifdim\stockwidth > \z@
2714         stockheight%
2715       \else
2716         paperheight%
2717       \fi
2718     }{%
2719       mediaheight%
2720     }%
2721   }{%
2722     \noexpand\ifcase\pdfpageheight
2723     \ifdim\stockwidth > \z@
2724       stockheight%
2725     \else
2726       paperheight%
2727     \fi
2728     \noexpand\else
2729     pdfpageheight%

```

```

2730     \noexpand\fi
2731   }%
2732 }
2733 \ifluatex
2734 \edef\ZREF@abspos@media@height{%
2735   \noexpand\ifcase\pageheight
2736     \ifdim\stockwidth > \z@
2737       stockheight%
2738     \else
2739       paperheight%
2740     \fi
2741   \noexpand\else
2742     pdfpageheight%
2743   \noexpand\fi}%
2744 \fi

```

(End of definition for \ZREF@abspos@media@height.)

`\ZREF@abspos@media@x@left`

```

2745 \def\ZREF@abspos@media@x@left#1{%
2746   0%
2747 }

```

(End of definition for \ZREF@abspos@media@x@left.)

`\ZREF@abspos@media@x@right`

```

2748 \def\ZREF@abspos@media@x@right#1{%
2749   \zref@extract{#1}\ZREF@abspos@media@width
2750 }

```

(End of definition for \ZREF@abspos@media@x@right.)

`\ZREF@abspos@media@x@center`

```

2751 \def\ZREF@abspos@media@x@center#1{%
2752   \ZREF@abspos@media@x@left{#1}%
2753   +\zref@extract{#1}\ZREF@abspos@media@width/2%
2754 }

```

(End of definition for \ZREF@abspos@media@x@center.)

`\ZREF@abspos@media@y@top`

```

2755 \def\ZREF@abspos@media@y@top#1{%
2756   \zref@extract{#1}\ZREF@abspos@media@height
2757 }

```

(End of definition for \ZREF@abspos@media@y@top.)

`\ZREF@abspos@media@y@bottom`

```

2758 \def\ZREF@abspos@media@y@bottom#1{%
2759   0%
2760 }

```

(End of definition for \ZREF@abspos@media@y@bottom.)

`\ZREF@abspos@media@y@center`

```

2761 \def\ZREF@abspos@media@y@center#1{%
2762   \zref@extract{#1}\ZREF@abspos@media@height/2%
2763 }

```

(End of definition for \ZREF@abspos@media@y@center.)

6.19.3 Paper

`\ZREF@abspos@paper@x@left`

```
2764 \def\ZREF@abspos@paper@x@left#1{%  
2765   0%  
2766 }
```

(End of definition for \ZREF@abspos@paper@x@left.)

`\ZREF@abspos@paper@x@right`

```
2767 \def\ZREF@abspos@paper@x@right#1{%  
2768   \zref@extract{#1}{paperwidth}%  
2769 }
```

(End of definition for \ZREF@abspos@paper@x@right.)

`\ZREF@abspos@paper@x@center`

```
2770 \def\ZREF@abspos@paper@x@center#1{%  
2771   \zref@extract{#1}{paperwidth}/2%  
2772 }
```

(End of definition for \ZREF@abspos@paper@x@center.)

`\ZREF@abspos@paper@y@top`

```
2773 \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top
```

(End of definition for \ZREF@abspos@paper@y@top.)

`\ZREF@abspos@paper@y@bottom`

```
2774 \def\ZREF@abspos@paper@y@bottom#1{%  
2775   \ZREF@abspos@paper@y@top{#1}%  
2776   -\zref@extract{#1}{paperheight}%  
2777 }
```

(End of definition for \ZREF@abspos@paper@y@bottom.)

`\ZREF@abspos@paper@y@center`

```
2778 \def\ZREF@abspos@paper@y@center#1{%  
2779   \ZREF@abspos@paper@y@top{#1}%  
2780   -\zref@extract{#1}{paperheight}/2%  
2781 }
```

(End of definition for \ZREF@abspos@paper@y@center.)

6.19.4 Origin

There doesn't seem a good reason to make these tests depend on pdf mode in current engines, so comment out the `\ifpdf` tests.

`\ZREF@abspos@origin@x`

```
2782 \let\ZREF@temp\ltx@two  
2783 \ltx@ifundefined{pdfhorigin}{-}{%  
2784 % \ifpdf  
2785   \let\ZREF@temp\ltx@zero  
2786 % \fi  
2787 }  
2788 \ifluatex  
2789 % \ifpdf  
2790   \let\ZREF@temp\ltx@zero  
2791 % \fi  
2792 \fi  
2793  
2794 \ifx\ZREF@temp\ltx@two  
2795   \ifnum\mag=1000 %  
2796     \let\ZREF@temp\ltx@one
```

```

2797 \fi
2798 \fi
2799 \ifcase\ZREF@temp
2800 \def\ZREF@abspos@origin@x#1{%
2801 \zref@extract{#1}{pdfhorigin}%
2802 }%
2803 \or
2804 \def\ZREF@abspos@origin@x#1{%
2805 4736286%
2806 }%
2807 \or
2808 \def\ZREF@abspos@origin@x#1{%
2809 \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2810 }%
2811 \fi

```

(End of definition for \ZREF@abspos@origin@x.)

\ZREF@abspos@origin@y

```

2812 \let\ZREF@temp\ltx@two
2813 \ltx@ifundefined{pdfvorigin}{}{%
2814 % \ifpdf
2815 \let\ZREF@temp\ltx@zero
2816 % \fi
2817 }
2818 \ifluatex
2819 % \ifpdf
2820 \let\ZREF@temp\ltx@zero
2821 % \fi
2822 \fi
2823 \ifx\ZREF@temp\ltx@two
2824 \ifnum\mag=1000 %
2825 \let\ZREF@temp\ltx@one
2826 \fi
2827 \fi
2828 \ifcase\ZREF@temp
2829 \def\ZREF@abspos@origin@y#1{%
2830 \zref@extract{#1}{pdfvorigin}%
2831 }%
2832 \or
2833 \def\ZREF@abspos@origin@y#1{%
2834 4736286%
2835 }%
2836 \or
2837 \def\ZREF@abspos@origin@y#1{%
2838 \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2839 }%
2840 \fi

```

(End of definition for \ZREF@abspos@origin@y.)

6.19.5 Header

\ZREF@abspos@head@x@left

```

2841 \def\ZREF@abspos@head@x@left#1{%
2842 \ZREF@abspos@paper@x@left{#1}%
2843 +\ZREF@abspos@origin@x{#1}%
2844 +\zref@extract{#1}{hoffset}%
2845 +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2846 \zref@extract{#1}{oddsidemargin}%
2847 \else
2848 \zref@extract{#1}{evensidemargin}%
2849 \fi
2850 }

```

(End of definition for \ZREF@abspos@head@x@left.)

\ZREF@abspos@head@x@right

```
2851 \def\ZREF@abspos@head@x@right#1{%
2852   \ZREF@abspos@head@x@left{#1}%
2853   +\zref@extract{#1}{textwidth}%
2854 }
```

(End of definition for \ZREF@abspos@head@x@right.)

\ZREF@abspos@head@x@center

```
2855 \def\ZREF@abspos@head@x@center#1{%
2856   \ZREF@abspos@head@x@left{#1}%
2857   +\zref@extract{#1}{textwidth}/2%
2858 }
```

(End of definition for \ZREF@abspos@head@x@center.)

\ZREF@abspos@head@y@top

```
2859 \def\ZREF@abspos@head@y@top#1{%
2860   \ZREF@abspos@paper@y@top{#1}%
2861   -\ZREF@abspos@origin@y{#1}%
2862   -\zref@extract{#1}{voffset}%
2863   -\zref@extract{#1}{topmargin}%
2864 }
```

(End of definition for \ZREF@abspos@head@y@top.)

\ZREF@abspos@head@y@bottom

```
2865 \def\ZREF@abspos@head@y@bottom#1{%
2866   \ZREF@abspos@head@y@top{#1}%
2867   -\zref@extract{#1}{headheight}%
2868 }
```

(End of definition for \ZREF@abspos@head@y@bottom.)

\ZREF@abspos@head@y@center

```
2869 \def\ZREF@abspos@head@y@center#1{%
2870   \ZREF@abspos@head@y@top{#1}%
2871   -\zref@extract{#1}{headheight}/2%
2872 }
```

(End of definition for \ZREF@abspos@head@y@center.)

6.19.6 Body

\ZREF@abspos@body@x@left

```
2873 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left
```

(End of definition for \ZREF@abspos@body@x@left.)

\ZREF@abspos@body@x@right

```
2874 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right
```

(End of definition for \ZREF@abspos@body@x@right.)

\ZREF@abspos@body@x@center

```
2875 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center
```

(End of definition for \ZREF@abspos@body@x@center.)

\ZREF@abspos@body@y@top

```
2876 \def\ZREF@abspos@body@y@top#1{%
2877   \ZREF@abspos@head@y@bottom{#1}%
2878   -\zref@extract{#1}{headsep}%
2879 }
```

(End of definition for \ZREF@abspos@body@y@top.)

\ZREF@abspos@body@y@bottom

```
2880 \def\ZREF@abspos@body@y@bottom#1{%
2881   \ZREF@abspos@body@y@top{#1}%
2882   -\zref@extract{#1}{textheight}%
2883 }
```

(End of definition for \ZREF@abspos@body@y@bottom.)

\ZREF@abspos@body@y@center

```
2884 \def\ZREF@abspos@body@y@center#1{%
2885   \ZREF@abspos@body@y@top{#1}%
2886   -\zref@extract{#1}{textheight}/2%
2887 }
```

(End of definition for \ZREF@abspos@body@y@center.)

6.19.7 Footer

\ZREF@abspos@foot@x@left

```
2888 \let\ZREF@abspos@foot@x@left\ZREF@abspos@head@x@left
```

(End of definition for \ZREF@abspos@foot@x@left.)

\ZREF@abspos@foot@x@right

```
2889 \let\ZREF@abspos@foot@x@right\ZREF@abspos@head@x@right
```

(End of definition for \ZREF@abspos@foot@x@right.)

\ZREF@abspos@foot@x@center

```
2890 \let\ZREF@abspos@foot@x@center\ZREF@abspos@head@x@center
```

(End of definition for \ZREF@abspos@foot@x@center.)

\ZREF@abspos@foot@y@bottom

```
2891 \def\ZREF@abspos@foot@y@bottom#1{%
2892   \ZREF@abspos@body@y@bottom{#1}%
2893   -\zref@extract{#1}{footskip}%
2894 }
```

(End of definition for \ZREF@abspos@foot@y@bottom.)

6.19.8 Marginal notes

\ZREF@abspos@marginpar@x@left

```
2895 \def\ZREF@abspos@marginpar@x@left#1{%
2896   \ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2897   \ZREF@abspos@body@x@right{#1}%
2898   +\zref@extract{#1}{marginparsep}%
2899   \else
2900   \ZREF@abspos@body@x@left{#1}%
2901   -\zref@extract{#1}{marginparsep}%
2902   -\zref@extract{#1}{marginparwidth}%
2903   \fi
2904 }
```

(End of definition for \ZREF@abspos@marginpar@x@left.)

\ZREF@abspos@marginpar@x@right

```
2905 \def\ZREF@abspos@marginpar@x@right#1{%
2906   \ZREF@abspos@marginpar@x@left{#1}%
2907   +\zref@extract{#1}{marginparwidth}%
2908 }
```

(End of definition for \ZREF@abspos@marginpar@x@right.)

\ZREF@abspos@marginpar@x@center

```
2909 \def\ZREF@abspos@marginpar@x@center#1{%  
2910   \ZREF@abspos@marginpar@x@left{#1}%  
2911   +\zref@extract{#1}{marginparwidth}/2%  
2912 }
```

(End of definition for \ZREF@abspos@marginpar@x@center.)

\ZREF@abspos@marginpar@y@top

```
2913 \let\ZREF@abspos@marginpar@y@top\ZREF@abspos@body@y@top
```

(End of definition for \ZREF@abspos@marginpar@y@top.)

\ZREF@abspos@marginpar@y@bottom

```
2914 \let\ZREF@abspos@marginpar@y@bottom\ZREF@abspos@body@y@bottom
```

(End of definition for \ZREF@abspos@marginpar@y@bottom.)

\ZREF@abspos@marginpar@y@center

```
2915 \let\ZREF@abspos@marginpar@y@center\ZREF@abspos@body@y@center
```

(End of definition for \ZREF@abspos@marginpar@y@center.)

6.19.9 Stock paper

\ZREF@abspos@stock@x@left

```
2916 \let\ZREF@abspos@stock@x@left\ZREF@abspos@paper@x@left
```

(End of definition for \ZREF@abspos@stock@x@left.)

\ZREF@abspos@stock@x@right

```
2917 \let\ZREF@abspos@stock@x@right\ZREF@abspos@paper@x@right
```

(End of definition for \ZREF@abspos@stock@x@right.)

\ZREF@abspos@stock@x@center

```
2918 \let\ZREF@abspos@stock@x@center\ZREF@abspos@paper@x@center
```

(End of definition for \ZREF@abspos@stock@x@center.)

\ZREF@abspos@stock@y@top

```
2919 \let\ZREF@abspos@stock@y@top\ZREF@abspos@paper@y@top
```

(End of definition for \ZREF@abspos@stock@y@top.)

\ZREF@abspos@stock@y@bottom

```
2920 \let\ZREF@abspos@stock@y@bottom\ZREF@abspos@paper@y@bottom
```

(End of definition for \ZREF@abspos@stock@y@bottom.)

\ZREF@abspos@stock@y@center

```
2921 \let\ZREF@abspos@stock@y@center\ZREF@abspos@paper@y@center
```

(End of definition for \ZREF@abspos@stock@y@center.)

```
2922 </abspos>
```

6.20 Module dotfill

```

2923 <*dotfill>
2924 \NeedsTeXFormat{LaTeX2e}
2925 \ProvidesPackage{zref-dotfill}%
2926 [2026-09-16 v2.38 Module dotfill for zref (HO)]%
2927 \RequirePackage{zref-base}[2019/11/29]
2928 \ifx\ZREF@base@ok Y%
2929 \else
2930 \expandafter\endinput
2931 \fi

```

For measuring the width of `\zdotfill` we use the features provided by module `savepos`.

```

2932 \RequirePackage{zref-savepos}[2019/11/29]

```

For automatically generated label names we use the unique counter of module `base`.

```

2933 \zref@require@unique

```

Configuration is done by the key value interface of package `keyval`.

```

2934 \RequirePackage{keyval}

```

The definitions of the keys follow.

```

2935 \define@key{ZREF@DF}{unit}{%
2936   \def\ZREF@df@unit{#1}%
2937 }
2938 \define@key{ZREF@DF}{min}{%
2939   \def\ZREF@df@min{#1}%
2940 }
2941 \define@key{ZREF@DF}{dot}{%
2942   \def\ZREF@df@dot{#1}%
2943 }

```

Defaults are set, see user interface.

```

2944 \providecommand\ZREF@df@min{2}
2945 \providecommand\ZREF@df@unit{.44em}
2946 \providecommand\ZREF@df@dot{.}

```

`\zdotfillsetup` Configuration of `\zdotfill` is done by `\zdotfillsetup`.

```

2947 \newcommand*{\zdotfillsetup}{\kvsetkeys{ZREF@DF}}

```

(End of definition for `\zdotfillsetup`. This function is documented on page 20.)

`\zdotfill` `\zdotfill` sets labels at the left and the right to get the horizontal position. `\zsavepos` is not used, because we do not need the vertical position.

```

2948 \ZREF@IfDefinable\zdotfill\def{%
2949   {%
2950     \leavevmode
2951     \global\advance\c@zref@unique\ltx@one
2952     \begingroup
2953     \def\ZREF@temp{zref@\number\c@zref@unique}%
2954     \csname tex_savepos:D\endcsname
2955     \zref@labelbyprops{\thezref@unique L}{posx}%
2956     \setlength{\dimen@}{\ZREF@df@unit}%
2957     \zref@ifrefundefined{\thezref@unique R}{%
2958       \ZREF@dotfill
2959     }{%
2960       \ifnum\numexpr\zposx{\thezref@unique R}%
2961         -\zposx{\thezref@unique L}\relax
2962         <\dimexpr\ZREF@df@min\dimen@\relax
2963       \hfill
2964     }%
2965     \ZREF@dotfill

```

```

2966         \fi
2967     }%
2968     \csname tex_savepos:D\endcsname
2969     \zref@labelbyprops{\thezref@unique R}{posx}%
2970 \endgroup
2971 \kern\z@
2972 }%
2973 }

```

(End of definition for `\zdotfill`. This function is documented on page 20.)

`\ZREF@dotfill` Help macro that actually sets the dots.

```

2974 \def\ZREF@dotfill{%
2975     \cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill
2976 }

```

(End of definition for `\ZREF@dotfill`.)

```

2977 </dotfill>

```

6.21 Module `env`

```

2978 <*env>
2979 \NeedsTeXFormat{LaTeX2e}
2980 \ProvidesPackage{zref-env}%
2981     [2026-09-16 v2.38 Module env for zref (HO)]%
2982 \RequirePackage{zref-base}[2019/11/29]
2983 \ifx\ZREF@base@ok Y%
2984 \else
2985     \expandafter\endinput
2986 \fi

```

```

2987 \zref@newprop{envname}[]{\@currentvir}
2988 \zref@newprop{envline}[]{\zref@env@line}

```

`\zref@env@line` Macro `\zref@env@line` extracts the line number from `\@currentvline`.

```

2989 \def\zref@env@line{%
2990     \ifx\@currentvline\ltx@empty
2991     \else
2992         \expandafter
2993         \ZREF@ENV@line\@currentvline\ltx@empty line \ltx@empty\@nil
2994     \fi
2995 }

```

(End of definition for `\zref@env@line`.)

`\ZREF@ENV@line`

```

2996 \def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%

```

(End of definition for `\ZREF@ENV@line`.)

```

2997 </env>

```

7 Installation

7.1 Download

Package. This package is available on CTAN²:

[CTAN:macros/latex/contrib/zref/zref.dtx](https://ctan.org/ctan/packages/macros/latex/contrib/zref/zref.dtx) The source file.

[CTAN:macros/latex/contrib/zref/zref.pdf](https://ctan.org/ctan/packages/macros/latex/contrib/zref/zref.pdf) Documentation.

²[CTAN:pkg/zref](https://ctan.org/ctan/packages/pkg/zref)

7.2 Package installation

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain $\mathrm{T}_{\mathrm{E}}\mathrm{X}$:

```
tex zref.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

<code>zref.sty</code>	<code>→ tex/latex/zref/zref.sty</code>
<code>zref-base.sty</code>	<code>→ tex/latex/zref/zref-base.sty</code>
<code>zref-abspage.sty</code>	<code>→ tex/latex/zref/zref-abspage.sty</code>
<code>zref-abspos.sty</code>	<code>→ tex/latex/zref/zref-abspos.sty</code>
<code>zref-counter.sty</code>	<code>→ tex/latex/zref/zref-counter.sty</code>
<code>zref-dotfill.sty</code>	<code>→ tex/latex/zref/zref-dotfill.sty</code>
<code>zref-env.sty</code>	<code>→ tex/latex/zref/zref-env.sty</code>
<code>zref-hyperref.sty</code>	<code>→ tex/latex/zref/zref-hyperref.sty</code>
<code>zref-lastpage.sty</code>	<code>→ tex/latex/zref/zref-lastpage.sty</code>
<code>zref-marks.sty</code>	<code>→ tex/latex/zref/zref-marks.sty</code>
<code>zref-nextpage.sty</code>	<code>→ tex/latex/zref/zref-nextpage.sty</code>
<code>zref-pageattr.sty</code>	<code>→ tex/latex/zref/zref-pageattr.sty</code>
<code>zref-pagelayout.sty</code>	<code>→ tex/latex/zref/zref-pagelayout.sty</code>
<code>zref-perpage.sty</code>	<code>→ tex/latex/zref/zref-perpage.sty</code>
<code>zref-runs.sty</code>	<code>→ tex/latex/zref/zref-runs.sty</code>
<code>zref-savepos.sty</code>	<code>→ tex/latex/zref/zref-savepos.sty</code>
<code>zref-thepage.sty</code>	<code>→ tex/latex/zref/zref-thepage.sty</code>
<code>zref-titleref.sty</code>	<code>→ tex/latex/zref/zref-titleref.sty</code>
<code>zref-totpages.sty</code>	<code>→ tex/latex/zref/zref-totpages.sty</code>
<code>zref-user.sty</code>	<code>→ tex/latex/zref/zref-user.sty</code>
<code>zref-xr.sty</code>	<code>→ tex/latex/zref/zref-xr.sty</code>
<code>zref.pdf</code>	<code>→ doc/latex/zref/zref.pdf</code>
<code>zref-example.tex</code>	<code>→ doc/latex/zref/zref-example.tex</code>
<code>zref-example-lastpage.tex</code>	<code>→ doc/latex/zref/zref-example-lastpage.tex</code>
<code>zref-example-nextpage.tex</code>	<code>→ doc/latex/zref/zref-example-nextpage.tex</code>
<code>zref.dtx</code>	<code>→ source/latex/zref/zref.dtx</code>

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

7.3 Refresh file name databases

If your $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ distribution ($\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Live, $\mathrm{MiK}_{\mathrm{T}}\mathrm{E}_{\mathrm{X}}$, ...) relies on file name databases, you must refresh these. For example, $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Live users run `texhash` or `mktextlsr`.

7.4 Some details for the interested

Unpacking with $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$. The `.dtx` chooses its action depending on the format:

plain $\mathrm{T}_{\mathrm{E}}\mathrm{X}$: Run docstrip and extract the files.

$\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$: Generate the documentation.

If you insist on using $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ for docstrip (really, docstrip does not need $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{zref.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL^AT_EX:

```
dflatex zref.dtx
akeindex -s gind.ist zref.idx
dflatex zref.dtx
akeindex -s gind.ist zref.idx
dflatex zref.dtx
```

8 References

- [1] Package `footmisc`, Robin Fairbairns, 2004/01/23 v5.3a. [CTAN:pkg/footmisc](#)
- [2] Package `hyperref`, Sebastian Rahtz, Heiko Oberdiek, 2006/08/16 v6.75c. [CTAN:pkg/hyperref](#)
- [3] Package `lastpage`, Jeff Goldberg, 1994/06/25 v0.1b. [CTAN:pkg/lastpage](#)
- [4] Package `nameref`, Sebastian Rahtz, Heiko Oberdiek, 2006/02/12 v2.24. [CTAN:pkg/nameref](#)
- [5] Package `perpage`, David Kastrup, 2002/12/20 v1.0. [CTAN:pkg/perpage](#)
- [6] Package `titleref`, Donald Arseneau, 2001/04/05 v3.1. [CTAN:pkg/titleref](#)
- [7] Package `totpages`, Wilhelm Müller, 1999/07/14 v1.00. [CTAN:pkg/totpages](#)
- [8] Package `xr`, David Carlisle, 1994/05/28 v5.02. [CTAN:pkg/xr](#)
- [9] Package `xr-hyper`, David Carlisle, 2000/03/22 v6.00beta4. [CTAN:pkg/xr-hyper](#)

9 History

[2006/02/20 v1.0]

- First version.

[2006/05/03 v1.1]

- Module `perpage` added.
- Module redesign as packages.

[2006/05/25 v1.2]

- Module `dotfillmin` added.
- Module `base`: macros `\zref@require@unique` and `\thezref@unique` added (used by modules `titleref` and `dotfillmin`).

[2006/09/08 v1.3]

- Typo fixes and English cleanup by Per Starback.

[2007/01/23 v1.4]

- Typo in macro name fixed in documentation.

[2007/02/18 v1.5]

- `\zref@getcurrent` added (suggestion of Igor Akkerman).
- Module `savepos` also supports $\text{\LaTeX}$.

[2007/04/06 v1.6]

- Fix in modules `abspage` and `base`: Now counter `abspage` and `zref@unique` are not remembered by `\include`.
- Beamer support for module `titleref`.

[2007/04/17 v1.7]

- Package `atbegshi` replaces `everyshi`.

[2007/04/22 v1.8]

- `\zref@wrapper@babel` and `\zref@refused` are now expandable if `babel` is not used or `\if@safe@actives` is already set to true. (Feature request of Josselin Noirel)

[2007/05/02 v1.9]

- Module `titleref`: Some support for `\caption` of package `longtable`, but only if `\label` is given after `\caption`.

[2007/05/06 v2.0]

- Uses package `etexcmds` for accessing $\text{\LaTeX}$'s `\unexpanded`.

[2007/05/28 v2.1]

- Module `titleref` supports caption of package `listings`.
- Fixes in module `titleref` for support of packages `titlesec` and `longtable`.

[2008/09/21 v2.2]

- Module `base`: `\zref@iflistcontainsprop` is documented, but a broken `\zref@listcontainsprop` implemented. Name and implementation fixed (thanks Ohad Kammar).

[2008/10/01 v2.3]

- `\zref@localaddprop` added (feature request of Ohad Kammar).
- Module `lastpage`: list ‘LastPage’ added. Label ‘LastPage’ will use the properties of this list (default is empty) along with the properties of the main list.

[2009/08/07 v2.4]

- Module `runs` added.

[2009/12/06 v2.5]

- Module `lastpage`: Uses package `atveryend`.
- Module `titleref`: Further commands are disabled during string expansion, imported from package `nameref`.

[2009/12/07 v2.6]

- Version date added for package atveryend.

[2009/12/08 v2.7]

- Module titleref: Use of package gettitlestring.

[2010/03/26 v2.8]

- `\zifrefundefined` added.
- Module lastpage: Macros `\zref@iflastpage` and `\ziflastpage` added.
- Module thepage added.
- Module nextpage added.

[2010/03/29 v2.9]

- Module marks added (without documentation).
- `\zref@addprop` now adds expanded property to list.
- Useless `\ZREF@ErrorNoLine` removed.

[2010/04/08 v2.10]

- Module xr remembers the external document name in property `'externaldocument'`.

[2010/04/15 v2.11]

- Module titleref: Better support of class memoir.
- Module titleref: Support of theorems.

[2010/04/17 v2.12]

- Module base: `\zref@newprop` ensures global empty default.
- Module xr: Setup options `tozreflabel` and `toltxlabel` added.

[2010/04/19 v2.13]

- `\zref@setcurrent` throws an error if the property does not exist (Florent Chervet).
- `\zref@getcurrent` the documentation is fixed (Florent Chervet). Also it returns the empty string in case of errors.
- `\zref@addprop` and `\zref@localaddprop` now take a list of property names (feature request of Florent Chervet).
- Example for `\zref@wrapper@unexpanded` corrected (Florent Chervet).

[2010/04/22 v2.14]

- Bug fix for `\zref@getcurrent` second argument wasn't eaten in case of unknown property.
- `\zref@getcurrent` supports `\zref@wrapper@unexpanded`.
- `\zref@wrapper@unexpanded` added for `\ZREF@xr@tolabel`.
- `\zref@extract`, `\zref@extractdefault`, `\zref@getcurrent` are expandable in exact two steps except inside `\zref@wrapper@unexpanded`.

[2010/04/23 v2.15]

- `\zexternaldocument` fixed for property 'url' when importing `\new@label` (bug found by Victor Ivrii).
- Two expansion steps also in `\zref@wrapper@unexpanded`.
- Nested calls of `\zref@wrapper@unexpanded` possible.

[2010/04/28 v2.16]

- More consequent use of package 'ltxcmds' and 'hologo'.
- Module `pagelayout` added.
- Module `pageattr` added.
- Robustness introduced for non-expandable interface macros.
- Internal change of the data format of property lists (suggestion of Florent Chervet).
- Module `titleref`: Support of environment description.

[2010/05/01 v2.17]

- `\zref@newprop` throws an error if the property already exists.
- Module `xr`: Bug fix for the case of several `.aux` files (bug found by Victor Ivrii).
- Module `xr`: Property 'urluse' and option `urluse` added.

[2010/05/13 v2.18]

- Module `env` added.
- Module `savepos`: `\zref@savepos` added.

[2010/10/22 v2.19]

- `\zref@addprop` and `\zref@localaddprop` are limited to one property only (incompatibility to versions v2.13 to v2.18).
- `\zref@addprops` and `\zref@localaddprops` added.
- `\zref@delprop` and `\zref@localdelprop` added.
- `\zref@labelbykv` and `\zkvlabel` (module `user`) with keys `prop`, `list`, `delprop`, `immediate`, `values` added.

[2011/02/12 v2.20]

- Fix for warning in `zref-xr`.

[2011/03/18 v2.21]

- Fix in module pagelayout for `\zlistpagelayout`.
- Fix for `\zref@localaddprop` (probably since v2.19).

[2011/10/05 v2.22]

- Documentation fixed for `\zref@localaddprop(s)`.
- Module base: `\zref@def@extract`, `\zref@def@extractdefault` added.
- Fix in module pagelayout: Because of missing `\noexpand` commands the values of the pagelayout properties on all pages were the values at package loading.
- Module base: `\zref@showprop` added.

[2011/12/05 v2.23]

- Module savepos: `\zsaveposx` and `\zsaveposy` added.

[2012/04/04 v2.24]

- Module titleref, package titlesec: some support for class ‘straight’ (`\ttl@straight@i`) added.

[2016/05/16 v2.25]

- Documentation updates.

[2016/05/21 v2.26]

- update zref-savepos for new luatex

[2018/11/21 v2.27]

- adapted zref-perpage, see issue <https://github.com/ho-tex/zref/issues/2>

[2019/11/29 v2.28]

- Documentation updates.
- Use `iftex` directly.

[2020-03-03 v2.29]

- adapted in module zref-pagelayout the properties `pdfhorigin`, `pdfvorigin`, `pdfpagewidth`, `pdfpageheight` for luatex to the right primitives.
- Removed no longer needed code for older luatex versions.
- added some documentation of the abspos module.
- adapted the abspos module to the new luatex primitives.
- adapted pageattr module to the new luatex primitives.
- added short documentation for pageattr module
- use luatex command names directly in zref-savepos rather than defining pdftex compatibility names.
- allow zref-abspos to use `\pdf[vh]origin` in dvi mode.

[2020-03-04 v2.30]

- add pagevalue property to savepos in the abspos module (issue 1)

[2020-05-28 v2.31]

- Adapted module zref-counter to use `\@currentcounter` in the next $\text{\LaTeX}$ version (issue 5)

[2020-07-03 v2.32]

- Changed in zref-pagelayout the names of the shipout box dimensions to adapt to the new hook management.

[2022-03-08 v2.33]

- Avoid that amstext undoes the stepcounter patch in zref-perpage, <https://github.com/ho-tex/zref/issues/11>
- Make the unique counter more robust when includeonly is used, <https://github.com/ho-tex/zref/issues/10>

[2022-04-07 v2.34]

- use the `zref@unique` counter in the include hook only if the module needs it <https://github.com/ho-tex/zref/issues/14>

[2023-09-14 v.2.35]

- Define the abspace counter only if it is undefined <https://github.com/ho-tex/zref/issues/14>
- Remove dependency from atveryend.

[2025-06-08 v2.36]

- use `\tex_savepos:D` for better engine compatibility, <https://github.com/ho-tex/zref/issues/20>
- adapt lastpage code to format change. <https://github.com/ho-tex/zref/issues/22>

[2026-03-07 v2.37]

- Updated `\zlabel` definition to new `\label` definition. <https://github.com/ho-tex/zref/issues/23>

[2026-09-16 v2.38]

- Adapted titleref module to `\DocumentMetadata`.
- Documentation is now tagged.
- Documented a number of missing commands.
- Removed `atbegshi` and removed `\zref@thepage@atbegshi@hook` in the zref-thepage module.

10 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

`\` 25, 26, 27, 28, 153, 155, 157, 158, 170,
173, 2344, 2438, 2463, 2473, 2477, 2493
`\_` 44, 45

A

`\AddLineBeginAux` 281
`\addtocounter` 19, 1758
`\AddToHook` 921, 1039, 1084, 1095,
1141, 1369, 1442, 1529, 1613, 1752
`\advance` 1098, 1425, 1581, 1806, 2157, 2951
`\afterassignment` 234, 1178, 1182
`\Alph` 7
`\anchor` 2475
`\AtBeginDocument` .. 1062, 1267, 1715, 1943
`\AtEndOfPackage` 195
`\autoref` 18, 46

B

`\begin` 23, 57, 100, 106, 156, 172, 1768
`\beginR` 19
`\bfseries` 937

C

`\caption` 12, 67, 92
`\chapter` 24, 30, 32, 61, 82
`\ChapterPages` 22, 91, 112
`\ChapterStart` 22, 78, 135, 150, 166
`\ChapterStop` 22, 85, 148, 165, 184
`\chardef` 1200, 1215, 1224, 1228
`\cleaders` 2975
`\cleardoublepage` 79, 86
`\clearpage` 62
`\closein` 2225
`\columnsep` 10, 17, 1339
`\columnwidth` 10, 17, 1338
count@ commands:
 `\count@:` 1386, 1572
`\cs` 1766
`\csname` 38, 253, 254, 291,
316, 317, 318, 327, 352, 353, 370,
371, 387, 388, 405, 406, 426, 428,
445, 463, 466, 479, 537, 539, 540,
545, 555, 560, 566, 579, 588, 605,
621, 652, 662, 735, 743, 749, 792,
793, 795, 815, 838, 839, 840, 901,
924, 1365, 1525, 1686, 1694, 1746,
1756, 1759, 1794, 1796, 1799, 1801,
1813, 1819, 1824, 1825, 1827, 1829,
1830, 1837, 1910, 1913, 2160, 2167,
2179, 2203, 2307, 2309, 2324, 2325,
2342, 2349, 2352, 2366, 2367, 2387,
2484, 2507, 2509, 2623, 2954, 2968
`\currentlabel` 7

D

`\DeclareBoolOption` 2136, 2137, 2138, 2142

`\DeclareOption` 194
`\default` 2456, 2496
`\descriptionlabel` 1987
`\detokenize` 64, 1876, 1894
`\dftest` 167, 174,
175, 176, 177, 178, 179, 180, 181, 182
dim commands:
 `\dim_use:N` 1347, 1348, 1349
`\dimexpr` .. 153, 155, 1437, 2809, 2838, 2962
`\do` 328, 725
`\documentclass` 4, 39, 68, 273
`\DocumentMetadata` 18, 63, 96
`\dotfill` 20, 24, 169, 173

E

`\edef` 11
`\emph` 150
`\end` 34, 64, 130, 159, 183, 185, 1763
`\endcsname` 253, 254,
291, 316, 317, 318, 327, 352, 353,
370, 371, 387, 388, 405, 406, 426,
428, 445, 464, 466, 479, 537, 539,
540, 545, 555, 560, 566, 579, 588,
605, 621, 652, 662, 741, 743, 749,
792, 793, 795, 815, 838, 839, 840,
888, 901, 924, 1365, 1525, 1686,
1694, 1745, 1746, 1755, 1756, 1759,
1771, 1794, 1796, 1799, 1800, 1801,
1813, 1819, 1824, 1825, 1827, 1829,
1830, 1837, 1910, 1913, 2160, 2167,
2179, 2203, 2307, 2309, 2324, 2325,
2342, 2349, 2352, 2366, 2367, 2387,
2484, 2507, 2510, 2623, 2954, 2968
`\endinginput` ... 192, 265, 278, 979, 1029,
1052, 1080, 1133, 1169, 1260, 1278,
1453, 1498, 1624, 1735, 1848, 2121,
2524, 2536, 2587, 2592, 2930, 2985
`\escapechar` 330, 421, 461, 462, 468, 727, 1284
`\evensidemargin` 10, 17, 1330
`\expandafter` 62
expandafter commands:
 `\expandafter:` 325
`\ExplSyntaxOff` 1350, 2547
`\ExplSyntaxOn` 1346, 2539
`\externaldocument` 21, 2311, 2353

F

`\fancyhead` 51, 54
`\fancyhf` 50, 53
`\fancypagestyle` 52
`\fmtversion` 223, 1054, 1083
`\foo` 18, 29, 31, 33
`\footskip` 10, 17, 20, 1335
`\foremargin` 1342
`\frontmatter` 58, 103

G	
<code>\gdef</code>	413, 540, 545, 957, 1365, 1722, 1794, 1796
<code>\GetTitleStringDisableCommands</code> ...	1905
<code>\GetTitleStringExpand</code>	1871, 1889
<code>\GetTitleStringNonExpand</code>	1873, 1891
<code>\GetTitleStringResult</code>	1876, 1894
<code>\glossary</code>	720
H	
<code>\headheight</code>	10, 17, 1333
<code>\headmargin</code>	1345
<code>\headsep</code>	10, 17, 20, 1334
<code>\hfill</code>	2963, 2975
<code>\hoffset</code>	10, 17, 1326
<code>\hss</code>	2975
I	
<code>\if</code>	32
<code>\ifcase</code>	115, 1236, 1657, 2722, 2735, 2799, 2828
<code>\ifcsname</code>	888, 1745, 1755, 1771, 1800
<code>\ifdim</code>	2695, 2713, 2723, 2736
<code>\IfDocumentMetadataT</code>	1885
<code>\IfDocumentMetadataTF</code>	954, 1918
<code>\ifeof</code>	2210, 2264
<code>\IfFileExists</code>	71
<code>\IfFormatAtLeastTF</code> ..	223, 919, 981, 1750
<code>\ifheadnameref</code>	2006, 2019
<code>\ifluatex</code>	1303, 1457, 1534, 1547, 1590, 1606, 2707, 2733, 2788, 2818
<code>\ifnum</code>	477, 1110, 1211, 1221, 1227, 1424, 1580, 1629, 1682, 2156, 2233, 2237, 2242, 2652, 2665, 2683, 2795, 2824, 2960
<code>\ifodd</code>	124, 2845, 2896
<code>\ifpdf</code>	2784, 2789, 2814, 2819
<code>\ifx</code>	438, 442, 475, 508, 566, 674, 677, 691, 734, 800, 977, 984, 993, 1001, 1027, 1050, 1078, 1131, 1167, 1258, 1276, 1364, 1451, 1497, 1505, 1524, 1535, 1591, 1622, 1639, 1648, 1652, 1657, 1658, 1659, 1733, 1824, 1846, 1998, 2002, 2059, 2079, 2103, 2119, 2167, 2250, 2273, 2278, 2283, 2297, 2344, 2435, 2438, 2463, 2466, 2473, 2477, 2493, 2522, 2534, 2585, 2590, 2794, 2823, 2928, 2983, 2990
<code>\immediate</code>	37, 706, 1721
<code>\include</code>	70, 92
<code>\index</code>	19, 63, 719
<code>\input</code>	70
int commands:	
<code>\int_use:N</code>	2540, 2541
<code>\item</code>	107, 111, 113, 121, 125, 127
K	
<code>\kern</code>	2971
<code>\kvsetkeys</code>	634, 1931, 2144, 2947
L	
<code>\label</code>	4–7, 11, 12, 19, 43–45, 63, 65, 67, 92, 718, 984, 993, 1001, 1939
<code>\leavevmode</code>	2950
M	
<code>\mag</code> ..	10, 17, 1296, 2795, 2809, 2824, 2838
<code>\mainmatter</code>	60, 134
<code>\makeatletter</code>	23, 11, 74, 101, 2180
<code>\makeatother</code>	16, 99
<code>\makebox</code>	169, 170
<code>\MakePerPage</code>	18, 61
<code>\MakeRobustcommand</code>	233
<code>\marginparsep</code>	10, 17, 1337
<code>\marginparwidth</code>	10, 17, 1336
<code>\meaning</code>	2100
<code>\mediaheight</code>	1302
<code>\mediawidth</code>	1301
<code>\MessageBreak</code>	271, 514, 565, 571, 681, 1374, 1386, 1390, 1438, 1548, 1573, 1576, 1607, 1609, 1632, 1635, 1662, 1664, 1703, 1704, 2162, 2163, 2212, 2229, 2234, 2238, 2243, 2299, 2418, 2427, 2657, 2669, 2670
N	
<code>\nameref</code>	18
<code>\NeedsTeXFormat</code>	3, 188, 220, 973, 1023, 1046, 1073, 1127, 1163, 1254, 1272, 1447, 1618, 1711, 1729, 1842, 2115, 2518, 2530, 2581, 2924, 2979
<code>\newcommand</code> ...	70, 18, 78, 85, 91, 167, 983, 992, 1000, 1006, 1177, 1194, 1195, 1264, 1626, 2143, 2572, 2575, 2597, 2604, 2611, 2615, 2664, 2947
<code>\newcount</code>	2146
<code>\newcounter</code>	6, 915, 1035, 1741
<code>\newdimen</code>	2690, 2691
<code>\newif</code>	248, 701, 1854, 1867, 2131
<code>\newlabel</code> 8, 21, 27, 70, 72, 2335, 2346, 2377	
<code>\newmarks</code>	1635
<code>\newpage</code>	143
<code>\noexpand</code>	95
<code>\nofiles</code>	17
<code>\number</code>	10, 94, 109, 917, 929, 967, 1145, 1289, 1290, 1305, 1306, 1309, 1310, 1313, 1314, 1317, 1318, 1716, 1788, 1814, 1821, 2598, 2600, 2605, 2607, 2612, 2616, 2666, 2678, 2845, 2896, 2953
<code>\numberwithin</code>	17
<code>\numexpr</code>	16, 19, 23, 94, 109, 115, 931, 1147, 1208, 1716, 1788, 2292, 2303, 2337, 2422, 2431, 2613, 2617, 2809, 2838, 2960
O	
<code>\oddsidemargin</code>	10, 17, 1329
<code>\openin</code>	2209
P	
<code>\PackageError</code>	259, 270, 295, 305, 490, 1630
<code>\PackageInfo</code>	292, 521, 1719, 2216, 2228, 2319, 2361, 2382
<code>\PackageWarning</code>	346, 364, 381, 399, 616, 680, 1660, 1701, 2211

<code>\@schapter</code>	1975	<code>\lst@@caption</code>	2082
<code>\@secondoftwo</code>	1460, 1538	<code>\lst@label</code>	2079
<code>\@sect</code>	1963	<code>\lst@MakeCaption</code>	2078
<code>\@spart</code>	1969	<code>\LT@caption</code>	2070
<code>\@ssect</code>	1981	<code>\ltx@backslashchar</code>	
<code>\@stpelt</code>	1759, 1769, 1770		748, 1546, 1605, 2162
<code>\@testdef</code>	1360, 1361, 1520, 1521	<code>\ltx@car</code>	1651, 2253
<code>\@testopt</code>	2183, 2186, 2196	<code>\ltx@cdr</code>	1653, 2256
<code>\@tfor</code>	323, 725	<code>\ltx@empty</code>	
<code>\@undefined</code>	1837, 2435		291, 500, 566, 633, 724, 801, 1139,
<code>\beamer@section</code>	2031		1639, 1855, 1881, 1899, 1908, 1998,
<code>\beamer@subsection</code>	2037		2002, 2059, 2079, 2192, 2227, 2250,
<code>\beamer@subsubsection</code>	2043		2256, 2297, 2451, 2990, 2993, 2996
<code>\c@abspage</code>	1814, 1821	<code>\ltx@firstofone</code>	
<code>\c@page</code>	967, 1098, 1811, 2845, 2896		255, 872, 883, 889, 1506, 1507
<code>\c@zpage</code>	1811, 1817	<code>\ltx@firstoftwo</code>	
<code>\c@zref@unique</code>			804, 831, 832, 897, 1112, 2685, 2687
.....	917, 924, 1806, 2951, 2953	<code>\ltx@firsttwo</code>	2676
<code>\ch@pt@c</code>	1998, 2004, 2010	<code>\ltx@gobble</code>	
<code>\comma@entry</code>	344, 345,		251, 356, 391, 624, 664, 985,
.....	347, 353, 379, 380, 382, 388, 615,		993, 994, 1001, 1427, 1583, 1670, 1939
.....	617, 621, 1657, 1658, 1659, 1665, 1668	<code>\ltx@gobblethree</code>	1002
<code>\comma@parse</code>	343, 378, 614, 1656	<code>\ltx@gobbletwo</code>	695, 1034, 1672, 1776
<code>\count@</code>	1375, 1387, 1389, 1424, 1425,	<code>\ltx@ifpackageloaded</code>	2404
.....	1434, 1436, 1437, 1549, 1574, 1575,	<code>\ltx@ifundefined</code>	
.....	1580, 1581, 2151, 2156, 2167, 2168		230, 250, 258, 411, 882,
<code>\current@chapid</code>	80, 88		928, 1144, 1472, 1484, 1503, 1504,
<code>\define@key</code>	1909, 1912, 1915,		1533, 1589, 1993, 1995, 2783, 2813
.....	1920, 1926, 2139, 2935, 2938, 2941	<code>\ltx@ifundefined</code>	
<code>\dimen@</code>	2956, 2962, 2975		301, 486, 584, 765, 810,
<code>\etex@unexpanded</code>	592, 824, 844, 2285		949, 1285, 1359, 1519, 2129, 2389,
<code>\filename@area</code>	2286		2390, 2620, 2693, 2694, 2711, 2712
<code>\filename@parse</code>	2195	<code>\ltx@LocalAppendToMacro</code>	
<code>\g@addto@macro</code>	351, 369, 1799		386, 404, 650, 660,
<code>\G@refundefinedtrue</code>	781		1603, 2308, 2348, 2351, 2456, 2459,
<code>\hb@xt@</code>	2975		2468, 2475, 2481, 2488, 2495, 2507
<code>\if@files</code>		<code>\ltx@newif</code>	1352, 1512
.....	711, 1085, 1096, 1720, 2544, 2550	<code>\ltx@one</code>	1425, 1581,
<code>\if@safe@actives</code>	92, 894		1806, 2151, 2157, 2796, 2825, 2951
<code>\ifetex@unexpanded</code>	268	<code>\ltx@onelevel@sanitize</code>	558, 563
<code>\iffirstchoice@</code>	1754	<code>\ltx@ReturnAfterFi</code>	2440
<code>\ifin@</code>	318	<code>\ltx@secondoftwo</code>	312,
<code>\ifZREF@found</code>	248, 2454, 2461		789, 802, 832, 891, 895, 1114, 2685
<code>\ifZREF@immediate</code>		<code>\ltx@space</code>	585, 587, 811, 820,
.....	37, 635, 701, 713, 721, 736		834, 837, 1221, 1227, 1388, 1435,
<code>\ifZREF@pa@list</code>	1512, 1517		1631, 1659, 1702, 2602, 2609, 2656
<code>\ifZREF@pl@list</code>	1352, 1357	<code>\ltx@two</code>	2782, 2794, 2812, 2823
<code>\ifzref@titleref@expand</code>		<code>\ltx@zero</code>	
.....	63, 1854, 1870, 1888		477, 1629, 1682, 1782, 2147, 2652,
<code>\ifzref@titleref@stripperperiod</code>			2665, 2683, 2785, 2790, 2815, 2820
.....	63, 1867, 1878, 1896	<code>\m@ne</code>	1098
<code>\ifZREF@xr@toltxlabel</code>	2330, 2372	<code>\M@sect</code>	2018
<code>\ifZREF@xr@tozreflabel</code>	2316, 2358	<code>\M@TitleReference</code>	2398
<code>\ifZREF@xr@urluse</code>	2200, 2479, 2509	<code>\new@label</code>	94
<code>\ifZREF@xr@verbose</code>	2318, 2360, 2381	<code>\nfss@text</code>	937
<code>\ifZREF@xr@zreflabel</code>		<code>\NR@temp</code>	2001, 2002
.....	69, 2131, 2217, 2231, 2272	<code>\on@line</code>	1718, 2227
<code>\in@</code>	315	<code>\pdf@escapehex</code>	1509
<code>\kv@define@key</code>	645, 656, 667, 672, 689	<code>\pdf@strcmp</code>	477
<code>\kv@key</code>	693, 1651, 1653, 1654, 1668	<code>\pdf@unescapehex</code>	1510
<code>\kv@parse</code>	690, 1647	<code>\protected@write</code>	716
<code>\kv@value</code>	691, 1648, 1649, 1656	<code>\ref@newprop</code>	43

<code>\ref@refused</code>	9	<code>\ZREF@abspos@marginpar@x@right</code>	2905
<code>\reset@font</code>	937	<code>\ZREF@abspos@marginpar@y@bottom</code>	2914
<code>\thezref@unique</code>	11, 42, 91, 916, 1809, 1810, 1817, 1818, 1820, 2955, 2957, 2960, 2961, 2969	<code>\ZREF@abspos@marginpar@y@center</code>	2915
<code>\toks@</code>		<code>\ZREF@abspos@marginpar@y@top</code> ..	2913
. 424, 430, 444, 445, 554, 557, 559,		<code>\ZREF@abspos@media@height</code>	2710, 2756, 2762
562, 613, 620, 621, 627, 745, 747,		<code>\ZREF@abspos@media@width</code>	2692, 2749, 2753
760, 761, 1087, 1091, 1099, 1104,		<code>\ZREF@abspos@media@x@center</code> ...	2751
1374, 1378, 1384, 1385, 1546, 1553,		<code>\ZREF@abspos@media@x@left</code>	2745, 2752
1570, 1571, 1858, 1864, 2271, 2286		<code>\ZREF@abspos@media@x@right</code> ...	2748
<code>\TR@TitleReference</code> .	2393, 2453, 2492	<code>\ZREF@abspos@media@y@bottom</code> ...	2758
<code>\ttl@sect@i</code>	2051	<code>\ZREF@abspos@media@y@center</code> ...	2761
<code>\ttl@straight@i</code>	95, 2057	<code>\ZREF@abspos@media@y@top</code> .	2755, 2773
<code>\XR@ext</code>	21, 2129	<code>\ZREF@abspos@origin@x</code>	2782, 2843
<code>\z@</code>	2695, 2713, 2723, 2736, 2971	<code>\ZREF@abspos@origin@y</code>	2812, 2861
<code>\Z@C@</code>	25, 33	<code>\ZREF@abspos@paper@x@center</code>	2770, 2918
<code>\Z@D@</code>	25, 33	<code>\ZREF@abspos@paper@x@left</code>	2764, 2842, 2916
<code>\Z@D@page</code>	1195	<code>\ZREF@abspos@paper@x@right</code>	2767, 2917
<code>\Z@E@</code>	25, 33	<code>\ZREF@abspos@paper@y@bottom</code>	2774, 2920
<code>\Z@L@</code>	25	<code>\ZREF@abspos@paper@y@center</code>	2778, 2921
<code>\Z@L@LastPage</code>	1089, 1101	<code>\ZREF@abspos@paper@y@top</code>	2773, 2775, 2779, 2860, 2919
<code>\Z@L@main</code>	1088, 1100	<code>\ZREF@abspos@stock@x@center</code> ...	2918
<code>\Z@L@ZREF@temp</code> .	633, 637, 640, 651, 661	<code>\ZREF@abspos@stock@x@left</code>	2916
<code>\Z@R@</code>	25, 27	<code>\ZREF@abspos@stock@x@right</code> ...	2917
<code>\Z@X@</code>	25, 33	<code>\ZREF@abspos@stock@y@bottom</code> ...	2920
<code>\ZREF@</code>	25	<code>\ZREF@abspos@stock@y@center</code> ...	2921
<code>\zref@</code>	5, 25	<code>\ZREF@abspos@stock@y@top</code>	2919
<code>\zref@(local)addprop(s)</code>	95	<code>\ZREF@abspos@used</code>	2642, 2644
<code>\ZREF@@@delprop</code> ...	435, 437, 472, 474	<code>\ZREF@absposnum</code>	2613, 2617, 2619
<code>\ZREF@@@newprop</code>	539, 543	<code>\zref@absposnumused</code>	2664
<code>\ZREF@@delprop</code>	425, 434, 448, 465, 471, 481	<code>\zref@absposnumx</code> ...	2599, 2611, 2633
<code>\ZREF@@extract</code>	813, 819	<code>\zref@absposnumy</code> ...	2606, 2615, 2636
<code>\ZREF@@makeperpage</code> .	1782, 1788, 1792	<code>\zref@absposused</code>	2641
<code>\ZREF@@newprop</code>	517, 528, 531, 535	<code>\zref@absposx</code>	2597, 2627
<code>\ZREF@@perpage@step</code>	1797, 1805	<code>\zref@absposy</code>	2604, 2630
<code>\ZREF@abspos@body@x@center</code> ...	2875	<code>\zref@addprop</code>	6, 29, 93, 94, 76, 360, 1043, 1057, 1061, 1137, 1140, 1291, 1307, 1311, 1315, 1319, 1466, 1468, 1481, 1493, 1668, 1853, 2527, 2595, 2596
<code>\ZREF@abspos@body@x@left</code> .	2873, 2900	<code>\zref@addprops</code>	6, 94, 15, 341, 946, 1351, 1740, 2542
<code>\ZREF@abspos@body@x@right</code>	2874, 2897	<code>\ZREF@addtoks</code>	759
<code>\ZREF@abspos@body@y@bottom</code>	2880, 2892, 2914	<code>\ZREF@base@ok</code>	970, 977, 1027, 1050, 1078, 1131, 1167, 1258, 1276, 1451, 1622, 1733, 1846, 2119, 2522, 2534, 2585, 2928, 2983
<code>\ZREF@abspos@body@y@center</code>	2884, 2915	<code>\ZREF@call</code>	1200, 1215, 1224, 1228, 1236
<code>\ZREF@abspos@body@y@top</code>	2876, 2881, 2885, 2913	<code>\ZREF@def@abspos</code>	2627, 2630, 2633, 2636, 2638
<code>\ZREF@abspos@foot@x@center</code> ...	2890	<code>\zref@def@absposnumx</code>	2632
<code>\ZREF@abspos@foot@x@left</code>	2888	<code>\ZREF@def@absposnumy</code>	2638
<code>\ZREF@abspos@foot@x@right</code>	2889	<code>\zref@def@absposnumy</code>	2635
<code>\ZREF@abspos@foot@y@bottom</code> ...	2891	<code>\zref@def@absposx</code>	2626
<code>\ZREF@abspos@head@x@center</code>	2855, 2875, 2890	<code>\zref@def@absposy</code>	2629
<code>\ZREF@abspos@head@x@left</code>	2841, 2852, 2856, 2873, 2888	<code>\ZREF@def@extract</code>	850, 852
<code>\ZREF@abspos@head@x@right</code>	2851, 2874, 2889		
<code>\ZREF@abspos@head@y@bottom</code>	2865, 2877		
<code>\ZREF@abspos@head@y@center</code> ...	2869		
<code>\ZREF@abspos@head@y@top</code>	2859, 2866, 2870		
<code>\ZREF@abspos@marginpar@x@center</code>	2909		
<code>\ZREF@abspos@marginpar@x@left</code> ..	2895, 2906, 2910		

\zref@def@extract 8, 95, 849
\ZREF@def@extractdefault .. 861, 863
\zref@def@extractdefault . 8, 95, 860
\ZREF@default 562, 563, 572
\zref@default . 9, 43, 531, 811, 934, 936
\ZREF@delprop
..... 413, 416, 418, 453, 456, 458
\zref@delprop 6, 94, 412, 452
\ZREF@df@dot 2942, 2946, 2975
\ZREF@df@min 2939, 2944, 2962
\ZREF@df@unit 2936, 2945, 2956
\ZREF@dotfill 2958, 2965, 2974
\ZREF@ENV@line 2993, 2996
\zref@env@line 89, 2988, 2989
\ZREF@ErrorNoLine 93
\zref@externaldocument 21
\ZREF@extract 808, 825, 828, 880
\zref@extract 8, 11, 40, 94,
95, 96, 109, 140, 808, 828, 857, 875,
880, 1013, 1150, 1246, 1389, 1436,
1437, 1560, 1597, 1817, 1818, 1940,
2573, 2576, 2749, 2753, 2756, 2762,
2768, 2771, 2776, 2780, 2801, 2830,
2844, 2846, 2848, 2853, 2857, 2862,
2863, 2867, 2871, 2878, 2882, 2886,
2893, 2898, 2901, 2902, 2907, 2911
\ZREF@extractdefault 829, 845, 848, 879
\zref@extractdefault 8,
11, 40, 94, 116, 117, 821, 848, 868,
874, 879, 1110, 1111, 1204, 1219,
1265, 1820, 2391, 2394, 2395, 2399,
2400, 2403, 2405, 2406, 2408, 2410,
2600, 2607, 2650, 2678, 2845, 2896
\ZREF@false 677, 687
\ZREF@foundfalse 2452
\ZREF@foundtrue 2499
\ZREF@getcurrent .. 582, 593, 596, 878
\zref@getcurrent
..... 7, 11, 34, 92-94, 596, 873, 878
\zref@hex
.. 1465, 1467, 1480, 1492, 1506, 1509
\zref@ifabsposnumundefined 2677, 2682
\zref@ifabsposundefined 2675
\ZREF@ifDefinable
. 242, 767, 1015, 1018, 1117, 1155,
1191, 1353, 1513, 1780, 1834, 1930,
1933, 2174, 2557, 2562, 2567, 2948
\ZREF@iflastpage 1118, 1120
\zref@iflastpage .. 13, 93, 1109, 1123
\zref@iflistcontainsprop 7,
28, 92, 310, 345, 363, 380, 398, 648, 658
\zref@iflistundefined
..... 6, 28, 289, 300, 304, 311
\zref@ifpropundefined
..... 7, 32, 485, 489,
519, 548, 615, 832, 1432, 1681, 2446
\ZREF@ifrefcontainsprop ... 791, 799
\zref@ifrefcontainsprop
... 9, 39, 787, 1434, 2407, 2503, 2504
\ZREF@ifrefundefined
..... 769, 772, 1201, 1212, 1222
\zref@ifrefundefined ... 9, 38, 39,
764, 774, 780, 788, 831, 1213, 1387,
1574, 1810, 2646, 2676, 2684, 2957
\ZREF@immediatefalse 678
\ZREF@immediatetrue 675, 704
\ZREF@label 37,
603, 627, 637, 640, 710, 1091, 1104
\zref@label 7, 8, 597, 987, 996
\zref@labelbykv 8, 94, 630, 1004
\zref@labelbylist
.... 7, 62, 598, 600, 1142, 1809, 2559
\zref@labelbyprops 8,
88, 610, 1199, 2564, 2569, 2955, 2969
\zref@listcontainsprop 92
\zref@listexists 6, 28, 303,
322, 342, 361, 377, 396, 419, 459, 602
\zref@listforloop 321, 657
\zref@localaddprop 6, 92-95, 395
\zref@localaddprops 6, 94, 376
\zref@localdelprop 6, 94, 415, 455, 669
\ZREF@mainlist 43, 598, 940,
943, 946, 1043, 1057, 1061, 1853, 2527
\ZREF@makeperpage@opt 1782, 1785
\ZREF@MARKS@DefineProp
..... 1644, 1645, 1646, 1680
\zref@marks@register
..... 1626, 1631, 1663, 1702
\ZREF@name 26, 229, 259, 270, 292, 295,
305, 346, 364, 381, 399, 490, 509,
521, 525, 549, 564, 616, 680, 692, 1630
\ZREF@NAME@bot 1659, 1679
\ZREF@NAME@first 1658, 1678
\ZREF@NAME@top 1657, 1677
\zref@newlabel 8,
21, 27, 70, 282, 285, 754, 2290, 2376
\zref@newlist 6, 28,
288, 943, 1082, 1136, 1654, 1739, 2538
\ZREF@newprop 498, 501, 504
\zref@newprop 7,
33, 93, 94, 12, 13, 14, 75, 495, 944,
945, 948, 955, 958, 963, 967, 1042,
1056, 1060, 1288, 1304, 1308, 1312,
1316, 1347, 1348, 1349, 1465, 1467,
1480, 1492, 1685, 1693, 2125, 2126,
2127, 2447, 2540, 2541, 2987, 2988
\ZREF@NewPropAnchor .. 947, 2176, 2526
\ZREF@NewPropPageValue 966, 1138, 1738
\ZREF@NewPropTheotype 962, 2347
\ZREF@NewPropTitle .. 953, 1852, 2177
\ZREF@nextpage 1192, 1196
\ZREF@nil 545, 801, 840, 2263,
2269, 2275, 2280, 2290, 2306, 2335,
2343, 2434, 2441, 2450, 2453, 2492
\ZREF@NOVALUE 807
\ZREF@novalue 800, 801, 807
\ZREF@np@call@next . 1186, 1190, 1245
\ZREF@np@call@nonext 1183, 1189, 1241
\ZREF@np@call@unknown 1179, 1188, 1237
\ZREF@np@setup@i 1178, 1181
\ZREF@np@setup@ii 1182, 1185
\ZREF@number 928, 1560, 1566, 1628, 2683
\ZREF@org@ 25

<code>\ZREF@org@@begintheorem</code>	2107	<code>\zref@propexists</code>	7, 32, 344, 362, 379, 397, 488, 578, 647, 668, 1007
<code>\ZREF@org@@caption</code>	1947	<code>\ZREF@refname@next</code>	1206, 1213, 1222, 1246
<code>\ZREF@org@@chapter</code>	1959, 2015	<code>\ZREF@refname@this</code>	1198, 1199, 1201, 1204
<code>\ZREF@org@@opargbegintheorem</code>	2092	<code>\ZREF@RefPrefix</code>	284, 286, 1364, 1524
<code>\ZREF@org@@part</code>	1953	<code>\ZREF@refused</code>	777, 779
<code>\ZREF@org@@schapter</code>	1977	<code>\zref@refused</code>	8, 13, 39, 92, 773, 776, 853, 864, 1012, 1019, 1121, 1122, 1153, 1268, 1566, 1601, 1938, 2645, 2653, 2666
<code>\ZREF@org@@sect</code>	1965	<code>\zref@require@unique</code>	11, 91, 913, 1742, 2933
<code>\ZREF@org@@spart</code>	1971	<code>\ZREF@Robust</code>	232, 238, 244, 285, 288, 303, 310, 341, 360, 376, 395, 412, 415, 452, 455, 488, 495, 547, 577, 597, 600, 610, 630, 702, 776, 849, 860, 871, 887, 913, 933, 939, 1152, 1565, 1600, 1856, 1869, 1887, 2626, 2629, 2632, 2635, 2641
<code>\ZREF@org@@ssect</code>	1983	<code>\ZREF@SavedEscapechar</code>	461, 468
<code>\ZREF@org@@stpelt</code>	1769, 1774, 1778	<code>\zref@savepos</code>	19, 94, 2543, 2551, 2553
<code>\ZREF@org@beamer@section</code>	2033	<code>\ZREF@savepos@ok</code>	2578, 2590
<code>\ZREF@org@beamer@subsection</code>	2039	<code>\zref@setcurrent</code>	7, 34, 93, 81, 541, 577, 697, 1065
<code>\ZREF@org@beamer@subsubsection</code>	2045	<code>\zref@setdefault</code>	7, 9, 43, 933, 936
<code>\ZREF@org@descriptionlabel</code>	1989	<code>\zref@setmainlist</code>	9, 43, 939
<code>\ZREF@org@lst@MakeCaption</code>	2085	<code>\zref@showprop</code>	6, 95, 547
<code>\ZREF@org@LT@c@option</code>	2071	<code>\ZREF@STAR</code>	1652, 1676
<code>\ZREF@org@M@sect</code>	2024	<code>\ZREF@stripperiod</code>	64, 1880, 1898, 1908
<code>\ZREF@org@refstepcounter</code>	1066	<code>\ZREF@temp</code>	62, 193, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 233, 234, 440, 441, 442, 724, 745, 746, 754, 1282, 1296, 1297, 1298, 1299, 1300, 1301, 1302, 1321, 1322, 1323, 1324, 1326, 1327, 1328, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1363, 1364, 1456, 1469, 1482, 1494, 1497, 1502, 1503, 1504, 1505, 1523, 1524, 1532, 1533, 1534, 1535, 1588, 1589, 1590, 1591, 1651, 1652, 2058, 2059, 2150, 2160, 2163, 2167, 2649, 2652, 2653, 2782, 2785, 2790, 2794, 2796, 2799, 2812, 2815, 2820, 2823, 2825, 2828, 2953
<code>\ZREF@org@stepcounter</code>	1743, 1748, 1773	<code>\ZREF@TempName</code>	1627, 1639, 1640, 1642, 1668, 1681, 1685, 1693, 1704
<code>\ZREF@org@testdef</code>	1360, 1362, 1520, 1522	<code>\ZREF@TempNum</code>	1628, 1629, 1633, 1640, 1682, 1695
<code>\ZREF@org@thepage</code>	714, 722	<code>\zref@thepage</code>	15, 1149, 1158
<code>\ZREF@org@ttl@sect@i</code>	2053	<code>\zref@thepage@atbegshi@hook</code>	96, 1139
<code>\ZREF@org@ttl@straight@i</code>	2064	<code>\zref@thepage@name</code>	14, 1144, 1150, 1153, 1207
<code>\ZREF@org@write</code>	705, 706	<code>\zref@thepage@refused</code>	15, 1152, 1157
<code>\ZREF@P</code>	505, 506, 508, 510, 519, 522, 526, 536, 537, 539, 540, 541, 545, 729, 730, 739, 743, 748, 749	<code>\ZREF@titleref</code>	1934, 1936
<code>\ZREF@pa@AfterLastShipout</code>	1516, 1614	<code>\zref@titleref@cleanup</code>	63, 1856, 1916
<code>\ZREF@pa@AtVeryEnd</code>	1529, 1532, 1603	<code>\zref@titleref@current</code>	44, 63, 953, 1875, 1879, 1880, 1893, 1897, 1898, 1902, 1921, 1922, 1927
<code>\ZREF@pa@ListPage</code>	1550, 1568		
<code>\ZREF@pa@listtrue</code>	1514		
<code>\ZREF@page@max</code>	1358, 1424, 1518, 1580		
<code>\zref@pageattr</code>	1563, 1564		
<code>\zref@pageattr@used</code>	1565		
<code>\ZREF@pagenum@last</code>	1218, 1221		
<code>\ZREF@pagenum@this</code>	1203, 1208, 1211, 1221, 1227		
<code>\ZREF@par</code>	508, 533		
<code>\ZREF@param</code>	422, 423, 442, 460, 477, 646, 647, 648, 652, 673, 674, 677, 682		
<code>\ZREF@patch</code>	27, 249, 1063, 1944, 1950, 1956, 1962, 1968, 1974, 1980, 1986, 2017, 2030, 2036, 2042, 2050, 2056, 2069, 2077, 2089, 2104		
<code>\zref@pdfpageattr</code>	1476, 1558, 1575		
<code>\zref@pdfpageattr@used</code>	1477		
<code>\zref@pdfpagesattr</code>	1488, 1595, 1608		
<code>\zref@pdfpagesattr@used</code>	1489, 1600		
<code>\ZREF@pl@AfterLastShipout</code>	1356, 1443		
<code>\ZREF@pl@AtVeryEnd</code>	1369, 1372		
<code>\ZREF@pl@ListEntry</code>	1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1419, 1420, 1431		
<code>\ZREF@pl@ListPage</code>	1376, 1382		
<code>\ZREF@pl@listtrue</code>	1354		
<code>\zref@pos@label@used</code>	2656		
<code>\zref@pos@num@used</code>	2670		
<code>\zref@prop</code>	324, 332, 333, 337, 658, 662		

\ZREF@titleref@hook	\ZREF@xr@refname
..... 63, 1855, 1859, 1863, 1906	.. 2291, 2317, 2324, 2336, 2359, 2366
\zref@titleref@setcurrent	\ZREF@xr@relax
..... 63, 1869, 1946,	2379, 2466
1952, 1958, 1964, 1970, 1976, 1982,	\ZREF@xr@scanparams
1988, 1996, 1999, 2003, 2007, 2009,	2341, 2450
2020, 2022, 2032, 2038, 2044, 2052,	\ZREF@xr@scantitleref
2060, 2062, 2072, 2081, 2091, 2106	2453, 2492
\zref@titleref@stripperiodtrue 1868	\ZREF@xr@temp
\ZREF@true	2465, 2466
674, 688	\ZREF@xr@tempname 2294, 2295, 2315,
\ZREF@u@getcurrent	2320, 2331, 2339, 2340, 2357, 2373
591	\ZREF@xr@temprefname
\zref@unhex 1507, 1510, 1559, 1596	 2295, 2307, 2309,
\ZREF@UpdatePdfTeX	2325, 2340, 2342, 2349, 2352, 2367
26, 247	\ZREF@xr@theURL
\ZREF@value	 2158, 2160, 2168, 2203, 2484
557, 558, 571	\ZREF@xr@tolabel 94, 2331, 2373, 2380
\ZREF@wrapper@babel	\ZREF@xr@URL ... 2146, 2156, 2157, 2158
904, 910	\ZREF@xr@url ... 2199, 2201, 2202, 2510
\zref@wrapper@babel	\ZREF@xr@urlcheck .. 2315, 2357, 2502
8,	\ZREF@xr@zref@ignorewarning
10, 92, 140, 769, 777, 850, 861,	 2327, 2369, 2416
887, 987, 996, 1004, 1008, 1118,	\ZREF@xr@zref@newlabel ... 2273, 2376
1934, 2627, 2630, 2633, 2636, 2642	\ZREF@xr@zreflabelfalse
\zref@wrapper@immediate	2182
... 11, 22, 37, 46, 87, 636, 702, 1103	\ZREF@xr@zreflabeltrue
\ZREF@wrapper@unexpanded .. 871, 885	2185
\zref@wrapper@unexpanded	\ZREF@zref
... 7, 11, 93, 94, 872, 877, 882, 2386	1008, 1011
\ZREF@wu@extract	\ZREF@zsavepos . 2548, 2559, 2564, 2569
823, 875	tex commands:
\ZREF@wu@extractdefault ... 843, 874	\tex_lastxpos:D
\ZREF@wu@getcurrent	2540
591, 873	\tex_lastypos:D
\ZREF@X	2541
497, 500, 537	\tex_savepos:D
\zref@xr@	2545
2140	\textheight
\ZREF@xr@@AddUrl	10, 17, 1332
2152, 2155	\textwidth
\ZREF@xr@input	10, 17, 1331
2283, 2378	\the .. 13, 153, 155, 430, 445, 461, 557,
\ZREF@xr@AddURL 2148, 2201, 2480	562, 621, 627, 747, 761, 931, 1042,
\ZREF@xr@checkfile . 2205, 2208, 2258	1091, 1104, 1142, 1147, 1208, 1358,
\ZREF@xr@checkkey	1378, 1385, 1386, 1387, 1389, 1434,
2436, 2445	1436, 1437, 1465, 1467, 1480, 1492,
\ZREF@xr@checklist	1518, 1553, 1571, 1572, 1574, 1864,
2306, 2434	2286, 2292, 2303, 2337, 2422, 2431
\zref@xr@ext	\thechapter
21, 2128, 2196	14
\ZREF@xr@externaldocument	\thefoo
..... 2183, 2186, 2190	7, 12, 20
\ZREF@xr@externalfile	\thetype
..... 2193, 2194, 2312, 2354	2349
\ZREF@xr@file ... 2194, 2209, 2212,	\thepage
2218, 2229, 2252, 2300, 2419, 2428	15,
\ZREF@xr@filelist	43, 44, 45, 714, 722, 783, 945, 1812
2192,	\thezpage
2250, 2253, 2255, 2256, 2284, 2285	18
\ZREF@xr@found . 2220, 2230, 2292, 2337	\thezpage
\ZREF@xr@graburl	18, 62, 1812, 1818
2196, 2198	\thezref@unique
\ZREF@xr@ignored@empty	11
..... 2221, 2233, 2235, 2302, 2303	\title
\ZREF@xr@ignored@ltx	2468, 2497
..... 2223, 2242, 2244, 2430, 2431	\titleref
\ZREF@xr@ignored@zref	18, 19
..... 2222, 2237, 2239, 2421, 2422	\topmargin
\ZREF@xr@line .. 2262, 2263, 2275, 2280	10, 17, 1328
\ZREF@xr@list	\trimedgedge
2296, 2297	1340
\ZREF@xr@ltx@ignorewarning ... 2425	\trimtop
\ZREF@xr@newlabel	1343
2278, 2377	
\ZREF@xr@prefix	
2191, 2291,	
2327, 2331, 2336, 2362, 2369, 2373	
\ZREF@xr@process@label ... 2280, 2335	
\ZREF@xr@process@zreflabel 2275, 2290	
\ZREF@xr@processfile	
2208, 2261	
\ZREF@xr@processline	
2263, 2269	

U

\unexpanded	11, 92, 271, 276
\UniqueCounterCall	1192
\UniqueCounterNew	1175
\uppermargin	1344
\url	2488
\urluse	2483
\usepackage	9, 41, 48, 70, 72

V

\value	13, 1142, 1358, 1518
\verb	173

\voffset	10, 17, 1327	\zref@extractdefault	8
W			
\write	37, 705, 706, 1721	\zref@getcurrent	7
X			
\x	331, 336, 728, 733, 902, 904, 1234, 1250, 1286, 1293, 1377, 1380, 1383, 1423, 1551, 1556, 1569, 1579, 1684, 1689, 1692, 1698, 1787, 1790, 2097, 2098, 2103, 2270, 2273, 2278, 2283, 2293, 2296, 2306, 2307, 2338, 2343	\zref@iflastpage	13
Y			
\y	2099, 2100	\zref@iflistcontainsprop	7
Z			
\z	5, 25, 44, 2100, 2101, 2103	\zref@iflistundefined	6
\zdotfill	20	\zref@ifpropundefined	7
\zdotfill	20, 88, 170, 173, 2948	\zref@ifrefcontainsprop	9
\zdotfillsetup	20	\zref@ifrefundefined	9
\zdotfillsetup	20, 88, 2947	\zref@label	7
\zexternaldocument	21	\zref@labelbykv	8
\zexternaldocument	21, 69, 94, 2174	\zref@labelbylist	7
\ziflastpage	13	\zref@labelbyprops	8
\ziflastpage	13, 93, 1117	\zref@listexists	6
\zifrefundefined	9	\zref@newlabel	8
\zifrefundefined	9, 93, 767	\zref@newlist	6
\zkvlabel	12	\zref@newprop	7
\zkvlabel	8, 12, 94, 1000	ZREF@P commands:	
\zlabel	12	\ZREF@P:	725
\zlabel	11, 12,	\zref@propexists	7
	18, 20, 45, 83, 104, 138, 146, 717, 981	\zref@refused	8
\zlistpageattr	1513	\zref@require@unique	11
\zlistpagelayout	17	\zref@savepos	19
\zlistpagelayout	17, 95, 1353	\zref@setcurrent	7
\zmakeperpage	18	\zref@setdefault	9
\zmakeperpage	18, 61, 62, 1780	\zref@setmainlist	9
\znextpage	15	\zref@showprop	6
\znextpage	15, 51, 54, 1191	\zref@thepage	15
\znextpagesetup	15	\zref@thepage@name	14
\znextpagesetup	15, 42, 1177	\zref@wrapper@babel	10
\znonextpagename	15, 46, 1194, 1242	\zref@wrapper@immediate	11
\zpageref	12	\zref@wrapper@unexpanded	11
\zpageref	12, 14, 45, 126, 1015	\zref@xr@ext	21
\zposx	19	ZREF@xr@theURL commands:	
\zposx	19, 78, 153, 2572, 2960, 2961	\ZREF@xr@theURL:	2162
\zposy	19	\zrefused	12
\zposy	19, 78, 155, 2572	\zrefused	12, 19,
\zref	12		24, 45, 78, 92, 93, 161, 162, 163, 1018
\zref	12, 45, 25, 26, 27, 28,	\zruns	17
	112, 114, 123, 128, 129, 139, 1006, 1016	\zruns	17, 1714
\zref@addprop	6	\zsavepos	19
\zref@addprops	6	\zsavepos	19, 20, 88, 157, 158, 2557
\zref@def@extract	8	\zsaveposx	19
\zref@default	9	\zsaveposx	19, 95, 2562
\zref@delprop	6	\zsaveposy	19, 95, 2567
\zref@extract	8	\zthepage	14
		\zthepage	14, 1155
		\ztitleref	18
		\ztitleref	18, 65, 1933
		\ztitlerefsetup	18
		\ztitlerefsetup	18, 64, 1909
		\ztotpages	16
		\ztotpages	16, 51, 124, 1264
		\zunknownnextpagename	15
		\zunknownnextpagename	15, 1195, 1238
		\zumakeperpage	18
		\zumakeperpage	18, 62, 1834
		\zxrsetup	21
		\zxrsetup	21, 2143