

The zref package

Heiko Oberdiek*

2020-03-03 v2.29

Abstract

Package `zref` tries to get rid of the restriction in \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 \LaTeX behaviour	4
1.2	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	11
2.9	Counter for unique names	11
3	User interface	11
3.1	Module <code>user</code>	11
3.2	Module <code>abspage</code>	12
3.3	Module <code>lastpage</code>	13
3.3.1	Tests for last page	13
3.3.2	Example	13
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

*Please report any issues at <https://github.com/ho-tex/zref/issues>

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
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>	22
4	ToDo	22
5	Example	22
6	Implementation	25
6.1	Package <code>zref</code>	25
6.1.1	Identification	25
6.1.2	Load basic module	25
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>	41
6.2.11	Unique counter support	42
6.2.12	Utilities	42
6.2.13	Setup	42
6.3	Module <code>user</code>	43
6.4	Module <code>abspage</code>	44
6.5	Module <code>counter</code>	45
6.6	Module <code>lastpage</code>	45
6.7	Module <code>thepage</code>	46
6.8	Module <code>nextpage</code>	47
6.9	Module <code>totpages</code>	49
6.10	Module <code>pagelayout</code>	49
6.10.1	Define layout properties	50
6.11	Module <code>pageattr</code>	53
6.12	Module <code>marks</code>	56
6.13	Module <code>runs</code>	58
6.14	Module <code>perpage</code>	58
6.15	Module <code>titleref</code>	61
6.15.1	Implementation	61
6.15.2	User interface	62
6.15.3	Patches for section and caption commands	63
6.15.4	Environment <code>description</code>	64
6.15.5	Class <code>memoir</code>	64
6.15.6	Class <code>beamer</code>	64
6.15.7	Package <code>titlesec</code>	65
6.15.8	Package <code>longtable</code>	65

6.15.9	Package listings	65
6.15.10	Theorems	66
6.16	Module <code>xr</code>	66
6.17	Module <code>hyperref</code>	74
6.18	Module <code>savepos</code>	75
6.18.1	Identification	75
6.18.2	Availability	75
6.18.3	Setup	75
6.18.4	User macros	76
6.19	Module <code>abspos</code>	77
6.19.1	Identification	77
6.19.2	Media	80
6.19.3	Paper	81
6.19.4	Origin	82
6.19.5	Header	83
6.19.6	Body	84
6.19.7	Footer	84
6.19.8	Marginal notes	85
6.19.9	Stock paper	85
6.20	Module <code>dotfill</code>	86
6.21	Module <code>env</code>	87
7	Installation	87
7.1	Download	87
7.2	Bundle installation	88
7.3	Package installation	88
7.4	Refresh file name databases	89
7.5	Some details for the interested	89
8	References	89
9	History	90
[2006/02/20 v1.0]		90
[2006/05/03 v1.1]		90
[2006/05/25 v1.2]		90
[2006/09/08 v1.3]		90
[2007/01/23 v1.4]		90
[2007/02/18 v1.5]		90
[2007/04/06 v1.6]		90
[2007/04/17 v1.7]		90
[2007/04/22 v1.8]		90
[2007/05/02 v1.9]		90
[2007/05/06 v2.0]		91
[2007/05/28 v2.1]		91
[2008/09/21 v2.2]		91
[2008/10/01 v2.3]		91
[2009/08/07 v2.4]		91
[2009/12/06 v2.5]		91
[2009/12/07 v2.6]		91
[2009/12/08 v2.7]		91
[2010/03/26 v2.8]		91
[2010/03/29 v2.9]		92
[2010/04/08 v2.10]		92
[2010/04/15 v2.11]		92

[2010/04/17 v2.12]	92
[2010/04/19 v2.13]	92
[2010/04/22 v2.14]	92
[2010/04/23 v2.15]	92
[2010/04/28 v2.16]	93
[2010/05/01 v2.17]	93
[2010/05/13 v2.18]	93
[2010/10/22 v2.19]	93
[2011/02/12 v2.20]	93
[2011/03/18 v2.21]	93
[2011/10/05 v2.22]	94
[2011/12/05 v2.23]	94
[2012/04/04 v2.24]	94
[2016/05/16 v2.25]	94
[2016/05/21 v2.26]	94
[2018/11/21 v2.27]	94
[2019/11/29 v2.28]	94
[2020-03-03 v2.29]	94

10 Index 95

1 Introduction

Standard L^AT_EX's reference system with `\label`, `\ref`, and `\pageref` supports two properties, the appearance of the counter that is last incremented by `\refstepcounter` and the page with the `\label` command.

Unhappily L^AT_EX does not provide an interface for adding another properties. Packages such as `hyperref`, `nameref`, or `titleref` are forced to use ugly hacks to extend the reference system. These ugly hacks are 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 and the page 7 in the reference. Internally the reference is a list with two entries:

```
\r@myref → {2.1}{7}
```

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

There are several tries to add new properties:

hyperref uses a list of five properties instead of the standard list with two entries. This causes many compatibility problems with L^AT_EX and other packages.

titleref stores its title data into the first entry in the list. L^AT_EX is happy because it does only see its list with two entries. The situation becomes more difficult, if more properties are added this way. Then the macros form a nested structure inside the first reference argument for the label. Expandable extractions will then become painful.

1.2 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_{\langle your key \rangle}_key{\langle some text \rangle}
```

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.

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.

<code>\zref@newlist {<listname>}</code>

Declares a new empty property list.

<code>\zref@addprop {<listname>} {<propname>}</code> <code>\zref@localaddprop {<listname>} {<propname>}</code>

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.

<code>\zref@addprops {<listname>} {<propname list>}</code> <code>\zref@localaddprops {<listname>} {<propname list>}</code>

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.

<code>\zref@listexists {<listname>} {<then>}</code>

Executes <then> if the property list <listname> exists or raise an error otherwise.

<code>\zref@iflistundefined^{exp} {<listname>} {<then>} {<else>}</code>

Executes <then> if the list exists or <else> otherwise.

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

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@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>`.

<code>\zref@ifrefcontainsprop^{exp} {<refname>} {<propname>} {<then>} {<else>}</code>

Test whether a reference provides a property.

2.6 Setup

<code>\zref@default</code>

Holds the global default for unknown values.

<code>\zref@setdefault {<value>}</code>

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.

<code>\zref@setmainlist {<value>}</code>
--

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>

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

2.8 Wrapper for advanced situations

`\zref@wrapper@babel {...} {\langle name \rangle}`

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.

`\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 suport.
<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.

`\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**

With the help of package `atbegshi` a new counter `abspage` with absolute page numbers is provided. For technical and historical reasons the counter itself 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` the first page will be page 1 as expected. Also a new property `abspage` is defined and added to the main property list. 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`.

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.

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

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`

`\znextpage`

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 `\zunknownnextpagename` 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.

`\zunknownnextpagename`
`\znonextpagename`

If the next page is not known or available, then `\znextpage` uses these name macros as default. `\zunknownnextpagename` 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.

`\znextpagesetup` `{\langle unknown\rangle}` `{\langle no next\rangle}` `{\langle next\rangle}`

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 `\zunknownnextpagename` 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,R0]{\znextpage}
52 \fancypagestyle{plain}{%
53   \fancyhf{}%
54   \fancyhead[LE,R0]{\znextpage}%
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 T_EX engine like pdfT_EX. 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 L^AT_EX runs since last `.aux` file creation and prints the number in the `.log` file.

`\zrunsexp`

Prints the the total number of L^AT_EX 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. L^AT_EX 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 L^AT_EX 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 reimplementation 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`.

`\ztitleref {refname}`^{babel}

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

<code>\ztitlerefsetup {key₁=value₁, key₂=value₂, ...}</code>
--

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

`title=<value>`

Sets the current title.

`stripperiod=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 pdfTeX provides (and XeTeX). pdfTeX is able to tell the current position on the page. The page position is not instantly known. First the page must be constructed by 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.

<code>\zsavepos {<refname>}</code>
--

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

<code>\zsaveposx {<refname>}</code>
<code>\zsaveposy {<refname>}</code>

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

<code>\zposx^{exp} {<refname>}</code>
<code>\zposy^{exp} {<refname>}</code>

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 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 L^AT_EX that the reference is actually used (the notifying is not expandable). Therefore you should mark the reference as used by `\zrefused`.

This module uses pdfTeX's `\pdfsavepos`, `\pdflastxpos`, and `\pdflastypos`. They are available in PDF mode and since version 1.40.0 also in DVI mode.

\zref@savepos

Macro `\zref@savepos` performs the first part of `\zsavepos` by calling `\pdfsavepos` (if `.aux` files are writable).

Thus `\zsavepos` is basically `\zref@savepos` followed by `\zref@labelbylist{<refname>}{savepos}`. If `\TeXeTstate` is detected and enabled, `\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>\pdfpageheigh</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`= $\langle count\ value \rangle$

If the actual number of dots are smaller than $\langle count\ value \rangle$, then the dots are suppressed. Default: 2.

`unit`= $\langle dimen\ value \rangle$

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

`dot`= $\langle value \rangle$

The dot itself is given by $\langle value \rangle$. 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 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`.

`\zexternaldocument * [$\langle prefix \rangle$] babel { $\langle external\ document \rangle$ } [$\langle url \rangle$]`

See `\zexternaldocument` for a description of this option. The found labels also get a property `externaldocument` that remembers $\langle external\ document \rangle$. 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 \LaTeX 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 \TeX 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}
```

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`, 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 pdfTeX's "savepos" feature. The position on the page is not known before the page is constructed and shipped out. Therefore the position is stored in references and are available for calculations in the next L^AT_EX 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 \zsavapos{firstcol}Hello&\zsavapos{secondcol}World\
158 \zsavapos{secondline}Second line&foobar\
159 \end{tabular}
160
```

With `\zrefused` L^AT_EX is notified, if the references are not yet available and L^AT_EX can generate the rerun hint.

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

Test for module `\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 </example>
```

6 Implementation

6.1 Package zref

6.1.1 Identification

```
187 <*package>
188 \NeedsTeXFormat{LaTeX2e}
189 \ProvidesPackage{zref}
190 [2020-03-03 v2.29 A new reference scheme for LaTeX (HO)]%
```

6.1.2 Load basic module

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

Abort package loading if 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 $\langle propname \rangle$.
`\Z@E@propname`: Extract function for property $\langle propname \rangle$.
`\Z@X@propname`: Information whether a property value for property $\langle propname \rangle$ is expanded immediately or at shipout time.
`\Z@C@propname`: Current value of the property $\langle propname \rangle$.
`\Z@R@labelname`: Data for reference $\langle labelname \rangle$.
`\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 [2020-03-03 v2.29 Module base for zref (H0)]%
```

6.2.3 Utilities

```

223 \RequirePackage{ltxcmds}[2010/12/02]
224 \RequirePackage{infwarerr}[2010/04/08]
225 \RequirePackage{kvsetkeys}[2010/03/01]
226 \RequirePackage{kvdefinekeys}[2010/03/01]
227 \RequirePackage{pdftexcmds}[2010/04/01]
```

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

```

228 \def\ZREF@name{zref}

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

`\ZREF@Robust`

```

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

236 }{%
```

`\ZREF@Robust`

```

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

240 }
```

`\ZREF@ifDefinable`

```

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

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

```

246 \def\ZREF@UpdatePdfTeX{Update pdfTeX.}

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

\ZREF@patch Macro \ZREF@patch first checks the existence of the command and safes it.
248 \def\ZREF@patch#1{%
249   \ltx@ifundefined{#1}{%
250     \ltx@gobble
251   }{%
252     \expandafter\let\csname ZREF@org@#1\expandafter\endcsname
253     \csname #1\endcsname
254     \ltx@firstofone
255   }%
256 }
```

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.

```

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

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

```

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

\ZREF@RefPrefix
283 \def\ZREF@RefPrefix{Z@R}
```

`\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.

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

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.

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

`\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.

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

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

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

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

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

```

318 }%
319 }

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

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

\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.
359 \ZREF@Robust\def\zref@addprop#1#2{%
360   \zref@listexists{#1}{%
361     \zref@propexists{#2}{%
362       \zref@iflistcontainsprop{#1}{#2}{%
363         \PackageWarning\ZREF@name{%
364           Property '#2' is already in list '#1'%
365         }%
366       }{%
367         \begingroup\expandafter\endgroup

```

```

368      \expandafter\g@addto@macro
369      \csname Z@L@#1\expandafter\endcsname
370      \expandafter{\csname#2\endcsname}%
371    }%
372  }%
373 }%
374 }

```

\zref@localaddprops

```

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

```

\zref@localaddprop

```

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

```

```

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

```

\zref@delprop

```

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

```

\zref@localdelprop

```

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

```

```

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

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

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

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

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

\ZREF@delprop
457 \def\ZREF@delprop#1#2#3{%
458   \zref@listexists{#2}{%
459     \def\ZREF@param{#3}%
460     \edef\ZREF@SavedEscapechar{\the\escapechar}%
461     \escapechar=-1 %
462     \expandafter#1\csname Z@L@#2%

```

```

463     \expandafter\expandafter\expandafter\endcsname{%
464     \expandafter\expandafter\expandafter\ZREF@@delprop
465     \csname Z@L@#2\endcsname!%
466     }%
467     \escapechar=\ZREF@SavedEscapechar\relax
468     }%
469 }%

```

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

```

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

```

`\ZREF@@@delprop`

```

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

```

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.

```

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

```

`\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.

```

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

```

`\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.


```

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

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

\ZREF@par
532 \def\ZREF@par{par}
533 \@onelevel@sanitize\ZREF@par

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

```

\zref@showprop

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

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

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

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

```
581 \def\ZREF@getcurrent#1{%
582   \romannumeral0%
583   \ltx@ifundefined{Z@C@#1}{%
584     \ltx@space
585   }{%
586     \expandafter\expandafter\expandafter\ltx@space
587     \csname Z@C@#1\endcsname
588   }%
589 }
```

\ZREF@u@getcurrent

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

\zref@getcurrent

```
595 \let\zref@getcurrent\ZREF@getcurrent
```

6.2.8 Reference generation

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

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

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

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

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

```
609 \ZREF@Robust\def\zref@labelbyprops#1#2{%
610   \@bsphack
611   \begingroup
612   \toks@{}%
613   \comma@parse{#2}{%
614     \zref@ifpropundefined\comma@entry{%
615       \PackageWarning\ZREF@name{%
616         Property ‘\comma@entry’ is not known%
617       }%
618     }{%
619       \toks@\expandafter{%
620         \the\expandafter\toks@\csname\comma@entry\endcsname
621       }%
622     }%
623     \ltx@gobble
624   }%
625   \expandafter\endgroup
626   \expandafter\ZREF@label\expandafter{\the\toks@}{#1}%
627   \@esphack
628 }
```

`\zref@labelbykv`

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

```

644 \kv@define@key{ZREF@LABEL}{prop}{%
645   \edef\ZREF@param{#1}%
646   \zref@propexists\ZREF@param{%
647     \zref@iflistcontainsprop{ZREF@temp}\ZREF@param}{}%
648     \begingroup\expandafter\endgroup
649     \expandafter\ltx@LocalAppendToMacro
650     \expandafter\Z@L@ZREF@temp
651     \expandafter{\csname\ZREF@param\endcsname}%
652   }%
653 }%
654 }
655 \kv@define@key{ZREF@LABEL}{list}{%
656   \zref@listforloop{#1}{%
657     \zref@iflistcontainsprop{ZREF@temp}\zref@prop}{}%
658     \begingroup\expandafter\endgroup
659     \expandafter\ltx@LocalAppendToMacro
660     \expandafter\Z@L@ZREF@temp
661     \expandafter{\csname\zref@prop\endcsname}%
662   }%
663   \ltx@gobble
664 }%
665 }
666 \kv@define@key{ZREF@LABEL}{delprop}{%
667   \zref@propexists{#1}{%
668     \zref@localdelprop{ZREF@temp}{#1}%
669   }%
670 }
671 \kv@define@key{ZREF@LABEL}{immediate}[true]{%
672   \edef\ZREF@param{#1}%
673   \ifx\ZREF@param\ZREF@true
674     \ZREF@immediatetrue
675   \else
676     \ifx\ZREF@param\ZREF@false
677       \ZREF@immediatefalse
678     \else
679       \PackageWarning\ZREF@name{%
680         Option 'immediate' expects 'true' or 'false'.\MessageBreak
681         Ignoring invalid value '\ZREF@param'%
682       }%
683     \fi
684   \fi
685 }

\ZREF@false

686 \def\ZREF@false{false}

\ZREF@true

687 \def\ZREF@true{true}

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

```

```

697     \fi
698   }%
699 }

```

`\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.

```

700 \newif\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`.

```

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

```

`\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.

```

709 \def\ZREF@label#1#2{%
710   \if@filesw
711     \begingroup
712       \ifZREF@immediate
713         \let\ZREF@org@thepage\thepage
714         \fi
715       \protected@write\@auxout{%
716         \ifZREF@immediate
717           \let\thepage\ZREF@org@thepage
718         \fi
719       \let\ZREF@temp\ltx@empty
720       \@tfor\ZREF@P:=#1\do{%
721         \begingroup
722           \escapechar=-1 %
723           \edef\x{\endgroup
724             \def\noexpand\ZREF@P{%
725               \expandafter\string\ZREF@P
726             }%
727           }%
728         \x
729         \expandafter\ifx
730           \csname
731             \ifZREF@immediate
732               relax%
733             \else
734               Z@X@\ZREF@P%
735             \fi
736           \endcsname
737         \noexpand
738         \expandafter\let\csname Z@C@\ZREF@P\endcsname\relax
739       \fi
740       \toks@\expandafter{\ZREF@temp}%

```

```

741         \edef\ZREF@temp{%
742             \the\toks@
743             \ltx@backslashchar\ZREF@P{%
744                 \expandafter\noexpand\csname Z@C@\ZREF@P\endcsname
745             }%
746         }%
747     }%
748     }{%
749         \string\zref@newlabel{#2}{\ZREF@temp}%
750     }%
751     \endgroup
752 \fi
753 }
754 \def\ZREF@addtoks#1{%
755     \toks@\expandafter\expandafter\expandafter{%
756         \expandafter\the\expandafter\toks@#1%
757     }%
758 }

```

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 \LaTeX 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 \pdfTeX that allows the detection of contexts. Then the shorthand can detect, if they are executed inside \csname and protect themselves automatically.

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

```

759 \def\zref@ifrefundefined#1{%
760     \ltx@ifundefined{Z@R@#1}%
761 }

```

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

```

762 \ZREF@IfDefinable\zifrefundefined\def{%
763     #1{%
764         \zref@wrapper@babel\ZREF@ifrefundefined{#1}%
765     }%
766 }

```

$\text{\ZREF@ifrefundefined}$

```

767 \def\ZREF@ifrefundefined#1{%
768     \zref@refused{#1}%
769     \zref@ifrefundefined{#1}%
770 }

```

\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 \LaTeX and a warning is given.

```

771 \ZREF@Robust\def\zref@refused#1{%
772   \zref@wrapper@babel\ZREF@refused{#1}%
773 }

\ZREF@refused
774 \def\ZREF@refused#1{%
775   \zref@ifrefundefined{#1}{%
776     \protect\G@refundefinedtrue
777     \@latex@warning{%
778       Reference ‘#1’ on page \thepage \space undefined%
779     }%
780   }{}%
781 }

\zref@ifrefcontainsprop \zref@ifrefcontainsprop looks, if the reference #1 has the property #2 and calls
then #3 and #4 otherwise.
782 \def\zref@ifrefcontainsprop#1#2{%
783   \zref@ifrefundefined{#1}{%
784     \ltx@secondoftwo
785   }{%
786     \expandafter\ZREF@ifrefcontainsprop
787     \csname Z@E@#2\expandafter\endcsname
788     \csname#2\expandafter\expandafter\expandafter\endcsname
789     \expandafter\expandafter\expandafter{%
790       \csname Z@R@#1\endcsname
791     }%
792   }%
793 }
794 \def\ZREF@ifrefcontainsprop#1#2#3{%
795   \expandafter\ifx\expandafter\ZREF@novalue
796   #1#3#2\ZREF@novalue\ZREF@nil\ltx@empty
797   \expandafter\ltx@secondoftwo
798   \else
799   \expandafter\ltx@firstoftwo
800   \fi
801 }
802 \def\ZREF@novalue{\ZREF@NOVALUE}

\zref@extract \zref@extract is an abbreviation for the case that the default of the property is
used as default value.
803 \def\ZREF@extract#1#2{%
804   \romannumeral0%
805   \ltx@ifundefined{Z@D@#2}{%
806     \expandafter\ltx@space\zref@default
807   }{%
808     \expandafter\expandafter\expandafter\ZREF@@extract
809     \expandafter\expandafter\expandafter{%
810       \csname Z@D@#2\endcsname
811     }{#1}{#2}%
812   }%
813 }

\ZREF@@extract
814 \def\ZREF@@extract#1#2#3{%
815   \expandafter\expandafter\expandafter\ltx@space
816   \zref@extractdefault{#2}{#3}{#1}%
817 }

```

```

\ZREF@wu@extract
818 \def\ZREF@wu@extract#1#2{%
819   \etex@unexpanded\expandafter\expandafter\expandafter{%
820     \ZREF@extract{#1}{#2}%
821   }%
822 }

\zref@extract
823 \let\zref@extract\ZREF@extract

\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.
824 \def\ZREF@extractdefault#1#2#3{%
825   \romannumeral0%
826   \zref@ifrefundefined{#1}\ltx@firstoftwo{%
827     \zref@ifpropundefined{#2}\ltx@firstoftwo\ltx@secondoftwo
828   }{%
829     \ltx@space
830     #3%
831   }{%
832     \expandafter\expandafter\expandafter\ltx@space
833     \csname Z@E@#2\expandafter\expandafter\expandafter\endcsname
834     \csname Z@R@#1\expandafter\endcsname
835     \csname#2\endcsname{#3}\ZREF@nil
836   }%
837 }

\ZREF@wu@extractdefault
838 \def\ZREF@wu@extractdefault#1#2#3{%
839   \etex@unexpanded\expandafter\expandafter\expandafter{%
840     \ZREF@extractdefault{#1}{#2}{#3}%
841   }%
842 }

\zref@extractdefault
843 \let\zref@extractdefault\ZREF@extractdefault

\zref@def@extract
844 \ZREF@Robust\def\zref@def@extract#1{%
845   \zref@wrapper@babel{\ZREF@def@extract{#1}}%
846 }

\ZREF@def@extract
847 \def\ZREF@def@extract#1#2#3{%
848   \zref@refused{#2}%
849   \expandafter\expandafter\expandafter\def
850   \expandafter\expandafter\expandafter#1%
851   \expandafter\expandafter\expandafter{%
852     \zref@extract{#2}{#3}%
853   }%
854 }

\zref@def@extractdefault
855 \ZREF@Robust\def\zref@def@extractdefault#1{%
856   \zref@wrapper@babel{\ZREF@def@extractdefault{#1}}%
857 }

```


\ZREF@def@extractdefault

```

858 \def\ZREF@def@extractdefault#1#2#3#4{%
859   \zref@refused{#2}%
860   \expandafter\expandafter\expandafter\def
861   \expandafter\expandafter\expandafter#1%
862   \expandafter\expandafter\expandafter{%
863     \zref@extractdefault{#2}{#3}{#4}%
864   }%
865 }

```

\ZREF@wrapper@unexpanded

```

866 \ZREF@Robust{\long\def}\ZREF@wrapper@unexpanded#1{%
867   \let\zref@wrapper@unexpanded\ltx@firstofone
868   \let\zref@getcurrent\ZREF@wu@getcurrent
869   \let\zref@extractdefault\ZREF@wu@extractdefault
870   \let\zref@extract\ZREF@wu@extract
871   #1%
872   \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded
873   \let\zref@getcurrent\ZREF@getcurrent
874   \let\zref@extractdefault\ZREF@extractdefault
875   \let\zref@extract\ZREF@extract
876 }

```

\zref@wrapper@unexpanded

```

877 \ltx@ifundefined{etex@unexpanded}{%
878   \let\zref@wrapper@unexpanded\ltx@firstofone
879 }{%
880   \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded
881 }

```

6.2.10 Compatibility with babel

\zref@wrapper@babel

```

882 \ZREF@Robust{\long\def}\zref@wrapper@babel#1#2{%
883   \ifcsname if@safe@actives\endcsname
884     \expandafter\ltx@firstofone
885   \else
886     \expandafter\ltx@secondoftwo
887   \fi
888   {%
889     \if@safe@actives
890       \expandafter\ltx@secondoftwo
891     \else
892       \expandafter\ltx@firstoftwo
893     \fi
894     {%
895       \begingroup
896         \csname @safe@activetrue\endcsname
897         \edef\x{#2}%
898         \expandafter\endgroup
899         \expandafter\ZREF@wrapper@babel\expandafter{\x}{#1}%
900       }%
901     }{%
902       #1{#2}%
903     }%
904 }
905 \long\def\ZREF@wrapper@babel#1#2{%
906   #2{#1}%

```

907 }

6.2.11 Unique counter support

`\zref@require@unique` Generate the counter `zref@unique` if the counter does not already exist.

```
908 \ZREF@Robust\def\zref@require@unique{%
909   \ifundefined{c@zref@unique}{%
910     \begingroup
911       \let\@addtoreset\ltx@gobbbletwo
912       \newcounter{zref@unique}%
913     \endgroup
```

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

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

6.2.12 Utilities

`\ZREF@number`

```
919 \ltx@ifundefined{numexpr}{%
920   \def\ZREF@number#1{\number#1}%
921 }{%
922   \def\ZREF@number#1{\the\numexpr(#1)\relax}%
923 }
```

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`.

```
924 \ZREF@Robust\def\zref@setdefault#1{%
925   \def\zref@default{#1}%
926 }
```

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

```
927 \zref@setdefault{%
928   \nfss@text{\reset@font\bfseries ??}%
929 }
```

Main property list.

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

```
930 \ZREF@Robust\def\zref@setmainlist#1{%
931   \def\ZREF@mainlist{#1}%
932 }
933 \zref@setmainlist{main}
```

Now we create the list.

```
934 \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 apperance of the latest counter that is incremented by `\refstepcounter`

`page` the apperance of the page counter

```
935 \zref@newprop{default}{\@currentlabel}
936 \zref@newprop*{page}{\thepage}
937 \zref@addprops\ZREF@mainlist{default,page}
```

Properties

`\ZREF@NewPropAnchor`

```
938 \def\ZREF@NewPropAnchor{%
939   \zref@newprop{anchor}{%
940     \ltx@ifundefined{@currentHref}{\@currentHref}%
941   }%
942   \global\let\ZREF@NewPropAnchor\relax
943 }
```

`\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`.

`\ZREF@NewPropTitle`

```
944 \def\ZREF@NewPropTitle{%
945   \gdef\zref@titleref@current{%
946     \zref@newprop{title}{\zref@titleref@current}%
947   \global\let\ZREF@NewPropTitle\relax
948 }
```

`\ZREF@NewPropTheotype`

```
949 \def\ZREF@NewPropTheotype{%
950   \zref@newprop{theotype}{}%
951   \global\let\ZREF@NewPropTheotype\relax
952 }
```

`\ZREF@NewPropPageValue`

```
953 \def\ZREF@NewPropPageValue{%
954   \zref@newprop*{pagevalue}[0]{\number\c@page}%
955   \global\let\ZREF@NewPropPageValue\relax
956 }
```

Mark successful loading

```
957 \let\ZREF@base@ok=Y
958 \end{base}
```

6.3 Module user

```
959 (*user)
960 \NeedsTeXFormat{LaTeX2e}
961 \ProvidesPackage{zref-user}%
962   [2020-03-03 v2.29 Module user for zref (H0)]%
963 \RequirePackage{zref-base}[2019/11/29]
964 \ifx\ZREF@base@ok Y%
965 \else
966   \expandafter\endinput
```

967 \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.

```
968 \newcommand*\zlabel{%
969   \ifx\label\ltx@gobble
970     \expandafter\ltx@gobble
971   \else
972     \expandafter\zref@wrapper@babel\expandafter\zref@label
973   \fi
974 }%
```

`\zkvlabel`

```
975 \newcommand*\zkvlabel}[1]{%
976   \ifx\label\ltx@gobble
977     \expandafter\ltx@gobblethree
978   \fi
979   \zref@wrapper@babel{\zref@labelbykv{#1}}%
980 }%
```

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

```
981 \newcommand*\zref}[2][default]{% robust because of optional argument
982   \zref@propexists{#1}%
983   \zref@wrapper@babel\ZREF@zref{#2}{#1}%
984 }%
985 }%
986 \def\ZREF@zref#1{%
987   \zref@refused{#1}%
988   \zref@extract{#1}%
989 }%
```

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

```
990 \ZREF@ifDefinable\zpageref\def{%
991   {\zref[page]}%
992 }
```

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

```
993 \ZREF@ifDefinable\zrefused\def{%
994   {\zref@refused}%
995 }
```

996 `\user`

6.4 Module `abspage`

```
997 (*abspage)
998 \NeedsTeXFormat{LaTeX2e}
999 \ProvidesPackage{zref-abspage}%
1000 [2020-03-03 v2.29 Module abspage for zref (HO)]%
1001 \RequirePackage{zref-base}[2019/11/29]
```

```

1002 \ifx\ZREF@base@ok Y%
1003 \else
1004   \expandafter\endinput
1005 \fi

```

Module `abspage` adds a new property `abspage` to the `main` property list for absolute page numbers. These are recorded by the help of package `atbegshi`.

```

1006 \RequirePackage{atbegshi}[2011/10/05]%

```

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.

```

1007 \begingroup
1008   \let\@addtoreset\ltx@gobbletwo
1009   \newcounter{abspage}%
1010 \endgroup
1011 \setcounter{abspage}{0}%
1012 \AtBeginShipout{%
1013   \stepcounter{abspage}%
1014 }%
1015 \zref@newprop*{abspage}[0]{\the\c@abspage}%
1016 \zref@addprop\ZREF@mainlist{abspage}%

```

Note that counter `abspage` shows the previous page during page processing. Before shipout the counter is 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.

```

1017 \end{abspage}

```

6.5 Module `counter`

```

1018 \*counter
1019 \NeedsTeXFormat{LaTeX2e}
1020 \ProvidesPackage{zref-counter}%
1021   [2020-03-03 v2.29 Module counter for zref (HO)]%
1022 \RequirePackage{zref-base}[2019/11/29]
1023 \ifx\ZREF@base@ok Y%
1024 \else
1025   \expandafter\endinput
1026 \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.

```

1027 \zref@newprop{counter}{}
1028 \zref@addprop\ZREF@mainlist{counter}

```

`\refstepcounter` is the central macro where we know which counter is responsible for the reference.

```

1029 \AtBeginDocument{%
1030   \ZREF@patch{refstepcounter}{%
1031     \def\refstepcounter#1{%
1032       \zref@setcurrent{counter}{#1}%
1033       \ZREF@org@refstepcounter{#1}%
1034     }%
1035   }%
1036 }
1037 \end{counter}

```

6.6 Module `lastpage`

```

1038 \*lastpage
1039 \NeedsTeXFormat{LaTeX2e}
1040 \ProvidesPackage{zref-lastpage}%

```

```

1041 [2020-03-03 v2.29 Module lastpage for zref (H0)]%
1042 \RequirePackage{zref-base}[2019/11/29]
1043 \RequirePackage{zref-abspage}[2019/11/29]
1044 \RequirePackage{atveryend}[2009/12/07]
1045 \ifx\ZREF@base@ok Y%
1046 \else
1047 \expandafter\endinput
1048 \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.

```

1049 \zref@newlist{LastPage}
1050 \AfterLastShipout{%
1051 \if@files
1052 \begingroup
1053 \advance\c@page\m@ne
1054 \toks@\expandafter\expandafter\expandafter{%
1055 \expandafter\Z@L@main
1056 \Z@L@LastPage
1057 }%
1058 \expandafter\zref@wrapper@immediate\expandafter{%
1059 \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
1060 }%
1061 \endgroup
1062 \fi
1063 }

```

\zref@iflastpage

```

1064 \def\zref@iflastpage#1{%
1065 \ifnum\zref@extractdefault{#1}{abspage}{-1}=%
1066 \zref@extractdefault{LastPage}{abspage}{-2} %
1067 \expandafter\ltx@firstoftwo
1068 \else
1069 \expandafter\ltx@secondoftwo
1070 \fi
1071 }

```

\ziflastpage

```

1072 \ZREF@ifDefinable\ziflastpage\def{%
1073 {\zref@wrapper@babel\ZREF@iflastpage}%
1074 }

```

ZREF@iflastpage

```

1075 \def\ZREF@iflastpage#1{%
1076 \zref@refused{LastPage}%
1077 \zref@refused{#1}%
1078 \zref@iflastpage{#1}%
1079 }

```

```

1080 \end{lastpage}

```

6.7 Module thepage

```

1081 \langle thepage \rangle
1082 \NeedsTeXFormat{LaTeX2e}
1083 \ProvidesPackage{zref-thepage}%
1084 [2020-03-03 v2.29 Module thepage for zref (H0)]%

```

```

1085 \RequirePackage{zref-base}[2019/11/29]
1086 \ifx\ZREF@base@ok Y%
1087 \else
1088   \expandafter\endinput
1089 \fi

1090 \RequirePackage{atbegshi}[2011/10/05]
1091 \RequirePackage{zref-abspage}[2019/11/29]

1092 \zref@newlist{thepage}
1093 \zref@addprop{thepage}{page}
1094 \ZREF@NewPropPageValue

\zref@thepage@atbegshi@hook

1095 \let\zref@thepage@atbegshi@hook\ltx@empty

1096 \zref@addprop{thepage}{pagevalue}
1097 \AtBeginShipout{%
1098   \AtBeginShipoutAddToBox{%
1099     \zref@thepage@atbegshi@hook
1100     \zref@labelbylist{thepage\the\value{abspage}}{thepage}%
1101   }%
1102 }

\zref@thepage@name

1103 \ltx@ifundefined{numexpr}{%
1104   \def\zref@thepage@name#1{thepage\number#1}%
1105 }{%
1106   \def\zref@thepage@name#1{thepage\the\numexpr#1}%
1107 }

\zref@thepage

1108 \def\zref@thepage#1{%
1109   \zref@extract{\zref@thepage@name{#1}}{page}%
1110 }%

\zref@thepage@refused

1111 \ZREF@Robust\def\zref@thepage@refused#1{%
1112   \zref@refused{\zref@thepage@name{#1}}%
1113 }%

\zthepage

1114 \ZREF@ifdefinable\zthepage\def{%
1115   #1{%
1116     \zref@thepage@refused{#1}%
1117     \zref@thepage{#1}%
1118   }%
1119 }

1120 </thepage>

```

6.8 Module nextpage

```

1121 (*nextpage)
1122 \NeedsTeXFormat{LaTeX2e}
1123 \ProvidesPackage{zref-nextpage}%
1124   [2020-03-03 v2.29 Module nextpage for zref (HO)]%
1125 \RequirePackage{zref-base}[2019/11/29]
1126 \ifx\ZREF@base@ok Y%

```

```

1127 \else
1128   \expandafter\endinput
1129 \fi

1130 \RequirePackage{zref-abspage}[2019/11/29]
1131 \RequirePackage{zref-thepage}[2019/11/29]
1132 \RequirePackage{zref-lastpage}[2019/11/29]
1133 \RequirePackage{uniquecounter}[2009/12/18]

1134 \UniqueCounterNew{znextpage}
1135
1136 \newcommand*{\znextpagesetup}{%
1137   \afterassignment\ZREF@np@setup@i
1138   \def\ZREF@np@call@unknown##1%
1139 }
1140 \def\ZREF@np@setup@i{%
1141   \afterassignment\ZREF@np@setup@ii
1142   \def\ZREF@np@call@nonext##1%
1143 }
1144 \def\ZREF@np@setup@ii{%
1145   \def\ZREF@np@call@next##1%
1146 }
1147 \def\ZREF@np@call@unknown#1{#1}
1148 \def\ZREF@np@call@nonext#1{#1}
1149 \def\ZREF@np@call@next#1{#1}
1150 \ZREF@ifDefinable\znextpage\def{%
1151   {\UniqueCounterCall{znextpage}{\ZREF@nextpage}}}%
1152 }%
1153 \newcommand*{\znonextpagename}{}
1154 \newcommand*{\zunknownnextpagename}{\Z@D@page}
1155 \def\ZREF@nextpage#1{%
1156   \begingroup
1157     \def\ZREF@refname@this{zref@np#1}%
1158     \zref@labelbyprops\ZREF@refname@this{abspage}%
1159     \chardef\ZREF@call=0 % unknown
1160     \ZREF@ifrefundefined\ZREF@refname@this{%
1161       }{%
1162         \edef\ZREF@pagenum@this{%
1163           \zref@extractdefault\ZREF@refname@this{abspage}{0}%
1164         }%
1165         \edef\ZREF@refname@next{%
1166           \zref@thepage@name%
1167           \the\numexpr\ZREF@pagenum@this+1%
1168         }%
1169       }%
1170       \ifnum\ZREF@pagenum@this>0 %
1171         \ZREF@ifrefundefined{LastPage}{%
1172           \zref@ifrefundefined\ZREF@refname@next{%
1173             }{%
1174               \chardef\ZREF@call=2 % next page
1175             }%
1176           }{%
1177             \edef\ZREF@pagenum@last{%
1178               \zref@extractdefault{LastPage}{abspage}{0}%
1179             }%
1180             \ifnum\ZREF@pagenum@this<\ZREF@pagenum@last\ltx@space
1181               \ZREF@ifrefundefined\ZREF@refname@next{%
1182                 }{%
1183                   \chardef\ZREF@call=2 % next page
1184                 }%

```



```

1185         \else
1186         \ifnum\ZREF@pagenum@this=\ZREF@pagenum@this\ltx@space
1187         \chardef\ZREF@call=1 % no next page
1188         \fi
1189         \fi
1190     }%
1191     \fi
1192 }%
1193 \edef\x{%
1194     \endgroup
1195     \ifcase\ZREF@call
1196     \noexpand\ZREF@np@call@unknown{%
1197     \noexpand\zunknownnextpagename
1198     }%
1199     \or
1200     \noexpand\ZREF@np@call@nonext{%
1201     \noexpand\znonextpagename
1202     }%
1203     \else
1204     \noexpand\ZREF@np@call@next{%
1205     \noexpand\zref@extract{\ZREF@refname@next}{page}%
1206     }%
1207     \fi
1208 }%
1209 \x
1210 }
1211 </nextpage>

```

6.9 Module totpages

```

1212 (*totpages)
1213 \NeedsTeXFormat{LaTeX2e}
1214 \ProvidesPackage{zref-totpages}%
1215 [2020-03-03 v2.29 Module totpages for zref (H0)]%
1216 \RequirePackage{zref-base}[2019/11/29]
1217 \ifx\ZREF@base@ok Y%
1218 \else
1219     \expandafter\endinput
1220 \fi

```

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

```

1221 \RequirePackage{zref-abspage}[2019/11/29]
1222 \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.

```

1223 \newcommand*{\ztotpages}{%
1224     \zref@extractdefault{LastPage}{abspage}{0}%
1225 }

```

Also we mark the reference `LastPage` as used:

```

1226 \AtBeginDocument{%
1227     \zref@refused{LastPage}%
1228 }
1229 </totpages>

```

6.10 Module pagelayout

```

1230 (*pagelayout)

```

```

1231 \NeedsTeXFormat{LaTeX2e}
1232 \ProvidesPackage{zref-pagelayout}%
1233 [2020-03-03 v2.29 Module pagelayout for zref (H0)]%
1234 \RequirePackage{zref-base}[2019/11/29]
1235 \ifx\ZREF@base@ok Y%
1236 \else
1237 \expandafter\endinput
1238 \fi

1239 \RequirePackage{zref-thepage}[2019/11/29]
1240 \RequirePackage{iftex}[2019/11/07]%
1241 \RequirePackage{atveryend}[2010/03/24]

```

6.10.1 Define layout properties

```

1242 \def\ZREF@temp#1{%
1243 \begingroup
1244 \escapechar=-1 %
1245 \ltx@ifundefined{\string#1}{\endgroup}{%
1246 \edef\x{%
1247 \endgroup
1248 \noexpand\zref@newprop*{\string#1}%
1249 [\noexpand\number\noexpand#1]% hash-ok
1250 {\noexpand\number\noexpand#1}%
1251 \noexpand\zref@addprop{thepage}{\string#1}%
1252 }%
1253 \x
1254 }%
1255 }

1256 \ZREF@temp\mag
1257 \ZREF@temp\paperwidth
1258 \ZREF@temp\paperheight
1259 \ZREF@temp\stockwidth % memoir.cls, crop.sty
1260 \ZREF@temp\stockheight % memoir.cls, crop.sty
1261 \ZREF@temp\mediawidth % VTeX
1262 \ZREF@temp\mediaheight % VTeX
1263 \ifluatex
1264 \zref@newprop*{pdfvorigin}%
1265 [\number\pdfvariable vorigin]% hash-ok
1266 {\number\pdfvariable vorigin}%
1267 \zref@addprop{thepage}{pdfvorigin}
1268 \zref@newprop*{pdfhorigin}%
1269 [\number\pdfvariable horigin]% hash-ok
1270 {\number\pdfvariable horigin}%
1271 \zref@addprop{thepage}{pdfhorigin}
1272 \zref@newprop*{pdfpageheight}%
1273 [\number\pageheight]% hash-ok
1274 {\number\pageheight}%
1275 \zref@addprop{thepage}{pdfpageheight}
1276 \zref@newprop*{pdfpagewidth}%
1277 [\number\pagewidth]% hash-ok
1278 {\number\pagewidth}%
1279 \zref@addprop{thepage}{pdfpagewidth}
1280 \else
1281 \ZREF@temp\pdfpagewidth
1282 \ZREF@temp\pdfpageheight
1283 \ZREF@temp\pdfhorigin
1284 \ZREF@temp\pdfvorigin
1285 \fi
1286 \ZREF@temp\hoffset

```

```

1287 \ZREF@temp\voffset
1288 \ZREF@temp\topmargin
1289 \ZREF@temp\oddsidemargin
1290 \ZREF@temp\evensidemargin
1291 \ZREF@temp\textwidth
1292 \ZREF@temp\textheight
1293 \ZREF@temp\headheight
1294 \ZREF@temp\headsep
1295 \ZREF@temp\footskip
1296 \ZREF@temp\marginparwidth
1297 \ZREF@temp\marginparsep
1298 \ZREF@temp\columnwidth
1299 \ZREF@temp\columnsep
1300 \ZREF@temp\trimedge % memoir.cls
1301 \ZREF@temp\spinemargin % memoir.cls
1302 \ZREF@temp\foremargin % memoir.cls
1303 \ZREF@temp\trintop % memoir.cls
1304 \ZREF@temp\uppermargin % memoir.cls
1305 \ZREF@temp\headmargin % memoir.cls

1306 \zref@newprop*{outputboxwd}[Opt]{\AtBeginShipoutBoxWidth}
1307 \zref@newprop*{outputboxht}[Opt]{\AtBeginShipoutBoxHeight}
1308 \zref@newprop*{outputboxdp}[Opt]{\AtBeginShipoutBoxDepth}
1309 \zref@addprops{thepage}{outputboxwd,outputboxht,outputboxdp}

\ifZREF@pl@list
1310 \ltx@newif\ifZREF@pl@list

\zref@listpagelayout
1311 \ZREF@IfDefinable\zlistpagelayout\def{%
1312   {\global\ZREF@pl@listtrue}%
1313 }

\ZREF@pl@AfterLastShipout
1314 \def\ZREF@pl@AfterLastShipout{%
1315   \ifZREF@pl@list
1316     \edef\ZREF@page@max{\the\value{abspage}}%
1317     \ltx@ifundefined{ZREF@org@testdef}{%
1318       \let\ZREF@org@testdef\@testdef
1319       \def\@testdef##1##2##3{%
1320         \ZREF@org@testdef{##1}{##2}{##3}%
1321       \def\ZREF@temp{##1}%
1322       \ifx\ZREF@temp\ZREF@RefPrefix
1323         \expandafter\gdef\cname##1@##2\endcname{##3}%
1324       \fi
1325     }%
1326   }{%
1327     \AtVeryEndDocument{\ZREF@pl@AtVeryEnd}%
1328   \fi
1329 }

\ZREF@pl@AtVeryEnd
1330 \def\ZREF@pl@AtVeryEnd{%
1331   \begingroup
1332   \toks@{Page layout parameters:\MessageBreak}%
1333   \count@=1 %
1334   \ZREF@pl@ListPage

```

```

1335 \edef\x{\endgroup
1336 \noexpand\@PackageInfoNoLine{zref-pagelayout}{\the\toks@}%
1337 }%
1338 \x
1339 }

```

\ZREF@pl@ListPage

```

1340 \def\ZREF@pl@ListPage{%
1341 \edef\x{%
1342 \toks@={%
1343 \the\toks@
1344 Page \the\count@:\noexpand\MessageBreak
1345 \zref@ifrefundefined{thepage\the\count@}{\the\count@}{%
1346 \ltx@space\ltx@space mag = %
1347 \zref@extract{thepage\the\count@}{mag}%
1348 \noexpand\MessageBreak
1349 \ZREF@pl@ListEntry{paperwidth}%
1350 \ZREF@pl@ListEntry{paperheight}%
1351 \ZREF@pl@ListEntry{stockwidth}%
1352 \ZREF@pl@ListEntry{stockheight}%
1353 \ZREF@pl@ListEntry{mediawidth}%
1354 \ZREF@pl@ListEntry{mediaheight}%
1355 \ZREF@pl@ListEntry{pdfpagewidth}%
1356 \ZREF@pl@ListEntry{pdfpageheight}%
1357 \ZREF@pl@ListEntry{pdfhorigin}%
1358 \ZREF@pl@ListEntry{pdfvorigin}%
1359 \ZREF@pl@ListEntry{hoffset}%
1360 \ZREF@pl@ListEntry{voffset}%
1361 \ZREF@pl@ListEntry{topmargin}%
1362 \ZREF@pl@ListEntry{oddsidemargin}%
1363 \ZREF@pl@ListEntry{evensidemargin}%
1364 \ZREF@pl@ListEntry{textwidth}%
1365 \ZREF@pl@ListEntry{textheight}%
1366 \ZREF@pl@ListEntry{headheight}%
1367 \ZREF@pl@ListEntry{headsep}%
1368 \ZREF@pl@ListEntry{footskip}%
1369 \ZREF@pl@ListEntry{marginparwidth}%
1370 \ZREF@pl@ListEntry{marginparsep}%
1371 \ZREF@pl@ListEntry{columnwidth}%
1372 \ZREF@pl@ListEntry{columnsep}%
1373 \ZREF@pl@ListEntry{trimedge}%
1374 \ZREF@pl@ListEntry{spinewidth}%
1375 \ZREF@pl@ListEntry{foremargin}%
1376 \ZREF@pl@ListEntry{trimtop}%
1377 \ZREF@pl@ListEntry{uppermargin}%
1378 \ZREF@pl@ListEntry{headmargin}%
1379 }%
1380 }%
1381 }\x
1382 \ifnum\ZREF@page@max>\count@
1383 \advance\count@ by\ltx@one
1384 \else
1385 \expandafter\ltx@gobble
1386 \fi
1387 \ZREF@pl@ListPage
1388 }

```

\ZREF@pl@ListEntry

```

1389 \def\ZREF@pl@ListEntry#1{%
1390   \zref@ifpropundefined{#1}{%
1391     }{%
1392       \zref@ifrefcontainsprop{thepage\the\count@}{#1}{%
1393         \ltx@space\ltx@space#1 = %
1394         \zref@extract{thepage\the\count@}{#1}sp = %
1395         \the\dimexpr\zref@extract{thepage\the\count@}{#1}sp\relax
1396         \noexpand\MessageBreak
1397       }{%
1398     }%
1399 }

1400 \AfterLastShipout{%
1401   \ZREF@pl@AfterLastShipout
1402 }

1403 \</pagelayout>

```

6.11 Module `pageattr`

```

1404 (*pageattr)
1405 \NeedsTeXFormat{LaTeX2e}
1406 \ProvidesPackage{zref-pageattr}%
1407   [2020-03-03 v2.29 Module pageattr for zref (HO)]%
1408 \RequirePackage{zref-base}[2019/11/29]
1409 \ifx\ZREF@base@ok Y%
1410 \else
1411   \expandafter\endinput
1412 \fi

1413 \RequirePackage{iftex}[2019/11/07]%

1414 \let\ZREF@temp=N%
1415 \ifluatex
1416   \expandafter\@firstoftwo
1417 \else
1418   \expandafter\@secondoftwo
1419 \fi

1420 {%luatex
1421   \RequirePackage{zref-thepage}[2019/11/29]
1422   \RequirePackage{zref-lastpage}[2019/11/29]%
1423   \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfvariable pageattr}}%
1424   \zref@addprop{thepage}{pdfpageattr}%
1425   \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfvariable pagesattr}}%
1426   \zref@addprop{LastPage}{pdfpagesattr}%
1427   \let\ZREF@temp=Y%
1428 }

1429 {%other
1430   \ltx@ifundefined{pdfpageattr}{%
1431     \@PackageInfoNoLine{zref-pageattr}{%
1432       \string\pdfpageattr\space is not available%
1433     }%
1434     \def\zref@pdfpageattr#1{%
1435       \def\zref@pdfpageattr@used#1{%
1436         }{%
1437           \RequirePackage{zref-thepage}[2019/11/29]%
1438           \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfpageattr}}%
1439           \zref@addprop{thepage}{pdfpageattr}%
1440           \let\ZREF@temp=Y%
1441         }

```

```

1442 \ltx@ifundefined{pdfpagesattr}{%
1443   \@PackageInfoNoLine{zref-pageattr}{%
1444     \string\pdfpagesattr\space is not available%
1445   }%
1446   \def\zref@pdfpagesattr{%
1447     \def\zref@pdfpagesattrused{%
1448     }{%
1449       \RequirePackage{zref-lastpage}[2019/11/29]%
1450       \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfpagesattr}}%
1451       \zref@addprop{LastPage}{pdfpagesattr}%
1452       \let\ZREF@temp=Y%
1453     }%
1454   }%

1455 \ifx\ZREF@temp N%
1456   \expandafter\endinput
1457 \fi

1458 \RequirePackage{zref-abspage}[2019/11/29]
1459 \RequirePackage{atveryend}[2010/03/24]
1460 \RequirePackage{pdftexcmds}[2010/04/01]
1461 \let\ZREF@temp=Y%
1462 \ltx@ifundefined{pdf@escapehex}{\let\ZREF@temp=N}{%
1463 \ltx@ifundefined{pdf@unescapehex}{\let\ZREF@temp=N}{%
1464 \ifx\ZREF@temp N%
1465   \let\zref@hex\ltx@firstofone
1466   \let\zref@unhex\ltx@firstofone
1467 \else
1468   \let\zref@hex\pdf@escapehex
1469   \let\zref@unhex\pdf@unescapehex
1470 \fi

\ifZREF@pa@list

1471 \ltx@newif\ifZREF@pa@list

\zref@listpageattr

1472 \ZREF@ifdefinable\zlistpageattr\def{%
1473   {\ZREF@pa@listtrue}%
1474 }

\ZREF@pa@AfterLastShipout

1475 \def\ZREF@pa@AfterLastShipout{%
1476   \ifZREF@pa@list
1477     \edef\ZREF@page@max{\the\value{abspage}}%
1478     \ltx@ifundefined{ZREF@org@testdef}{%
1479       \let\ZREF@org@testdef\@testdef
1480       \def\@testdef##1##2##3{%
1481         \ZREF@org@testdef{##1}{##2}{##3}%
1482         \def\ZREF@temp{##1}%
1483         \ifx\ZREF@temp\ZREF@RefPrefix
1484           \expandafter\xdef\csname##1##2\endcsname{##3}%
1485         \fi
1486       }%
1487     }{%
1488       \AtVeryEndDocument{\ZREF@pa@AtVeryEnd}%
1489     \fi
1490 }

\ZREF@pa@AtVeryEnd

```

```

1491 \let\ZREF@temp=Y%
1492 \ltx@ifundefined{pdfpageattr}{\let\ZREF@temp=N}
1493 \ifluatex \let\ZREF@temp=N \fi
1494 \ifx\ZREF@temp Y
1495 \expandafter\@firstoftwo
1496 \else
1497 \expandafter\@secondoftwo
1498 \fi
1499 {%
1500 \def\ZREF@pa@AtVeryEnd{%
1501 }
1502 {%
1503 \def\ZREF@pa@AtVeryEnd{
1504 \begingroup
1505 \toks@{\List of \ltx@backslashchar
1506 \ifluatex pdfvariable\else pdf\fi
1507 pdfpageattr:\MessageBreak}%
1508 \count@=1 %
1509 \ZREF@pa@ListPage
1510 \edef\x{\endgroup
1511 \noexpand\@PackageInfoNoLine{zref-pageattr}{%
1512 \the\toks@
1513 }%
1514 }%
1515 \x
1516 }%

\zref@pageattr

1517 \def\zref@pdfpageattr#1{%
1518 \zref@unhex{%
1519 \zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}%
1520 }%
1521 }
1522 % compability, \zref@pageattr was defined in older versions
1523 \let\zref@pageattr\zref@pdfpageattr

```

\zref@pageattr@used

```

1524 \ZREF@Robust\def\zref@pageattr@used#1{%
1525 \zref@refused{thepage\ZREF@number{#1}}%
1526 }

```

\ZREF@pa@ListPage

```

1527 \def\ZREF@pa@ListPage{%
1528 \edef\x{%
1529 \toks@={%
1530 \the\toks@
1531 Page \the\count@%
1532 \noexpand\MessageBreak
1533 \zref@ifrefundefined{thepage\the\count@}{\fi
1534 <<\zref@pdfpageattr\count@>>%
1535 \noexpand\MessageBreak
1536 }%
1537 }%
1538 }\x
1539 \ifnum\ZREF@page@max>\count@
1540 \advance\count@ by\ltx@one
1541 \else
1542 \expandafter\ltx@gobble

```

```

1543 \fi
1544 \ZREF@pa@ListPage
1545 }%
1546 }

1547 \let\ZREF@temp=Y%
1548 \ltx@ifundefined{pdfpagesattr}{\let\ZREF@temp=N}
1549 \ifluatex \let\ZREF@temp=N \fi
1550 \ifx\ZREF@temp N
1551 \expandafter\@firstofone
1552 \fi
1553 {%

```

\zref@pdfpagesattr

```

1554 \def\zref@pdfpagesattr{%
1555 \zref@unhex{%
1556 \zref@extract{LastPage}{pdfpagesattr}%
1557 }%
1558 }%

```

\zref@pdfpagesattr@used

```

1559 \ZREF@Robust\def\zref@pdfpagesattr@used{%
1560 \zref@refused{LastPage}%
1561 }%

1562 \ltx@LocalAppendToMacro\ZREF@pa@AtVeryEnd{%
1563 \@PackageInfoNoLine{zref-pageattr}{%
1564 \ltx@backslashchar
1565 \ifluatex pdfvariable\else pdf\fi
1566 pagesattr:\MessageBreak
1567 <<\zref@pdfpagesattr>>%
1568 \MessageBreak
1569 }%
1570 }%
1571 }
1572 \AfterLastShipout{%
1573 \ZREF@pa@AfterLastShipout
1574 }
1575 \pageattr)

```

6.12 Module marks

```

1576 (*marks)
1577 \NeedsTeXFormat{LaTeX2e}
1578 \ProvidesPackage{zref-marks}%
1579 [2020-03-03 v2.29 Module marks for zref (H0)]%
1580 \RequirePackage{zref-base}[2019/11/29]
1581 \ifx\ZREF@base@ok Y%
1582 \else
1583 \expandafter\endinput
1584 \fi

1585 \newcommand*{\zref@marks@register}[3][{}]{%
1586 \edef\ZREF@TempName{#1}%
1587 \edef\ZREF@TempNum{\ZREF@number{#2}}%
1588 \ifnum\ZREF@TempNum<\ltx@zero %
1589 \PackageError\ZREF@name{%
1590 \string\zref@marks@register\ltx@space is called with invalid%
1591 \MessageBreak

```



```

1592     marks register number (\ZREF@TempNum)%
1593 }{%
1594     Use '0' or the command, defined by \string\newmarks.\MessageBreak
1595     \@ehc
1596 }%
1597 \else
1598     \ifx\ZREF@TempName\ltx@empty
1599         \edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}%
1600     \else
1601         \edef\ZREF@TempName{marks\ZREF@TempName}%
1602     \fi
1603     \ZREF@MARKS@DefineProp{top}%
1604     \ZREF@MARKS@DefineProp{first}%
1605     \ZREF@MARKS@DefineProp{bot}%
1606     \kv@parse{#3}{%
1607         \ifx\kv@value\relax
1608             \def\kv@value{top,first,bot}%
1609         \fi
1610         \edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}%
1611         \ifx\ZREF@temp\ZREF@STAR
1612             \edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}%
1613             \zref@newlist\kv@key
1614         \fi
1615         \expandafter\comma@parse\expandafter{\kv@value}{%
1616             \ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else
1617                 \ifx\comma@entry\ZREF@NAME@first 1\else
1618                     \ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space
1619             \PackageWarning{zref-marks}{%
1620                 Use 'top', 'first' or 'bot' for the list values%
1621                 \MessageBreak
1622                 in the third argument of \string\zref@marks@register.%
1623                 \MessageBreak
1624                 Ignoring unknown value '\comma@entry'%
1625             }%
1626         \else
1627             \zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}%
1628         \fi
1629         \ltx@gobble
1630     }%
1631     \ltx@gobbletwo
1632 }%
1633 \fi
1634 }
1635 \def\ZREF@STAR{*}
1636 \def\ZREF@NAME@top{top}
1637 \def\ZREF@NAME@first{first}
1638 \def\ZREF@NAME@bot{bot}
1639 \def\ZREF@MARKS@DefineProp#1{%
1640     \zref@ifpropundefined{#1\ZREF@TempName}{%
1641         \ifnum\ZREF@TempNum=\ltx@zero
1642             \begingroup
1643             \edef\x{\endgroup
1644                 \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
1645                     \expandafter\noexpand\csname#1mark\endcsname
1646                 }%
1647             }%
1648             \x
1649         \else

```

```

1650     \begingroup
1651     \edef\x{\endgroup
1652         \noexpand\zref@newprop*{#1\ZREF@TempName}[] {%
1653             \expandafter\noexpand\csname#1marks\endcsname
1654             \ZREF@TempNum
1655         }%
1656     }%
1657     \x
1658     \fi
1659 }{%
1660     \PackageWarning{zref-marks}{%
1661         \string\zref@marks@register\ltx@space does not generate the%
1662         \MessageBreak
1663         new property ‘#1\ZREF@TempName’, because\MessageBreak
1664         it is already defined%
1665     }%
1666 }%
1667 }
1668 </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.

```

1669 <*runs>
1670 \NeedsTeXFormat{LaTeX2e}
1671 \ProvidesPackage{zref-runs}%
1672 [2020-03-03 v2.29 Module runs for zref (H0)]%

```

\zruns

```

1673 \providecommand*{\zruns}{0}%
1674 \AtBeginDocument{%
1675     \edef\zruns{\number\numexpr\zruns+1}%
1676     \begingroup
1677     \def\on@line{}%
1678     \PackageInfo{zref-runs}{LaTeX runs: \zruns}%
1679     \if@filesw
1680         \immediate\write\@mainaux{%
1681             \string\gdef\string\zruns{\zruns}%
1682         }%
1683     \fi
1684     \endgroup
1685 }
1686 </runs>

```

6.14 Module perpage

```

1687 <*perpage>
1688 \NeedsTeXFormat{LaTeX2e}
1689 \ProvidesPackage{zref-perpage}%
1690 [2020-03-03 v2.29 Module perpage for zref (H0)]%
1691 \RequirePackage{zref-base}[2019/11/29]
1692 \ifx\ZREF@base@ok Y%
1693 \else
1694     \expandafter\endinput
1695 \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.

```
1696 \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.

```
1697 \ZREF@NewPropPageValue
1698 \zref@newlist{perpage}
1699 \zref@addprops{perpage}{abspage,page,pagevalue}
```

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

```
1700 \newcounter{zpage}
```

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

```
1701 \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`.

```
1702 \let\ZREF@org@stepcounter\stepcounter
1703 \def\stepcounter#1{%
1704   \ifcsname @stepcounterhook@#1\endcsname
1705     \csname @stepcounterhook@#1\endcsname
1706   \fi
1707   \ZREF@org@stepcounter{#1}%
1708 }
```

`\@stpelt` must be adapted due to the change in latex 2015-01, see <https://github.com/hotex/zref/issues/26>

```
1709 \let\ZREF@org@@stpelt\@stpelt
1710 \def\@stpelt#1{%
1711   \ifcsname ZREF@perpage@#1\endcsname
1712     \begingroup
1713       \let\stepcounter\ZREF@org@stepcounter
1714       \ZREF@org@@stpelt{#1}%
1715     \endgroup
1716     \expandafter\ltx@gobbletwo
1717   \fi
1718   \ZREF@org@@stpelt{#1}%
1719 }
```

`\zmakeperpage` Makro `\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.

```
1720 \ZREF@IfDefinable\zmakeperpage\def{%
1721   {%
1722     \@ifnextchar[\ZREF@makeperpage@opt{\ZREF@makeperpage[\ltx@zero]}%
1723   }%
1724 }
```

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

```
1725 \def\ZREF@makeperpage@opt[#1]{%
1726   \begingroup
1727     \edef\x{\endgroup
1728       \noexpand\ZREF@makeperpage[\number\numexpr#1-1\relax]%
1729     }%
```

```

1730 \x
1731 }
1732 \def\ZREF@makeperpage[#1]#2{%
1733 \ifundefined{@stepcounterhook@#2}{%
1734 \expandafter\gdef\csname @stepcounterhook@#2\endcsname{%
1735 }}%
1736 \expandafter\gdef\csname ZREF@perpage@#2\endcsname{%
1737 \ZREF@perpage@step{#2}{#1}%
1738 }%
1739 \expandafter\g@addto@macro\csname @stepcounterhook@#2\endcsname{%
1740 \ifcsname ZREF@perpage@#2\endcsname
1741 \csname ZREF@perpage@#2\endcsname
1742 \fi
1743 }%
1744 }

```

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

```

1745 \def\ZREF@perpage@step#1#2{%
First the reference is generated.
1746 \global\advance\c@zref@unique\ltx@one
1747 \begingroup
1748 \expandafter
1749 \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.

```

1750 \zref@ifredefined\thezref@unique{%
1751 \global\c@zpage=\c@page
1752 \global\let\thezpage\thepage
1753 \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1754 \number\c@abspage
1755 }%
1756 }%

```

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

```

1757 \global\c@zpage=\zref@extract\thezref@unique{pagevalue}\relax
1758 \xdef\thezpage{\noexpand\zref@extract{\thezref@unique}{page}}%
1759 \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1760 \zref@extractdefault\thezref@unique
1761 {abspage}{\number\c@abspage}%
1762 }%
1763 }%

```

Page changes are detected by a changed absolute page number.

```

1764 \expandafter\ifx\csname ZREF@abspage@#1\endcsname
1765 \csname ZREF@currentabspage@#1\endcsname
1766 \else
1767 \global\csname c@#1\endcsname=#2\relax
1768 \global\expandafter\let
1769 \csname ZREF@currentabspage@#1\endcsname
1770 \csname ZREF@abspage@#1\endcsname
1771 \fi
1772 \endgroup
1773 }

```

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

```

1774 \ZREF@ifDefinable\zunmakeperpage\def{%
1775   #1{%
1776     \global\expandafter
1777     \let\csname ZREF@perpage@#1\endcsname\@undefined
1778   }%
1779 }

1780 </perpage>

```

6.15 Module titleref

```

1781 (*titleref)
1782 \NeedsTeXFormat{LaTeX2e}
1783 \ProvidesPackage{zref-titleref}%
1784   [2020-03-03 v2.29 Module titleref for zref (HO)]%
1785 \RequirePackage{zref-base}[2019/11/29]
1786 \ifx\ZREF@base@ok Y%
1787 \else
1788   \expandafter\endinput
1789 \fi
1790 \RequirePackage{getttitlestring}[2009/12/08]

```

6.15.1 Implementation

```

1791 \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.

```

1792 \ZREF@NewPropTitle
1793 \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. 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.

```
\ifzref@titleref@expand
```

```

1794 \newif\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`.

```

1795 \let\ZREF@titleref@hook\ltx@empty

```

```
\zref@titleref@cleanup
```

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

```

1796 \ZREF@Robust\def\zref@titleref@cleanup#1{%
1797   \begingroup
1798   \toks@\expandafter{%
1799     \ZREF@titleref@hook
1800     #1%
1801   }%
1802   \expandafter\endgroup

```

```

1803 \expandafter\def\expandafter\ZREF@titleref@hook\expandafter{%
1804 \the\toks@
1805 }%
1806 }%

\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.
1807 \newif\ifzref@titleref@stripperperiod
1808 \zref@titleref@stripperperiodtrue

\zref@titleref@setcurrent Macro \zref@titleref@setcurrent sets a new current title stored in
\zref@titleref@current. Some cleanup and expansion is performed that can
be controlled by the previous switches.
1809 \ZREF@Robust\def\zref@titleref@setcurrent#1{%
1810 \ifzref@titleref@expand
1811 \GetTitleStringExpand{#1}%
1812 \else
1813 \GetTitleStringNonExpand{#1}%
1814 \fi
1815 \edef\zref@titleref@current{%
1816 \detokenize\expandafter{\GetTitleStringResult}}%
1817 }%
1818 \ifzref@titleref@stripperperiod
1819 \edef\zref@titleref@current{%
1820 \expandafter\ZREF@stripperperiod\zref@titleref@current
1821 \ltx@empty.\ltx@empty\@nil
1822 }%
1823 \fi
1824 }%
1825 \GetTitleStringDisableCommands{%
1826 \ZREF@titleref@hook
1827 }

\ZREF@stripperperiod If \ZREF@stripperperiod is called, the argument consists of space tokens and tokens
with catcode 12 (other), because of  $\varepsilon$ -TEX's \detokenize.
1828 \def\ZREF@stripperperiod#1.\ltx@empty#2\@nil{#1}%

```

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.
1829 \define@key{ZREF@TR}{expand}[true]{%
1830 \csname zref@titleref@expand#1\endcsname
1831 }%
1832 \define@key{ZREF@TR}{stripperperiod}[true]{%
1833 \csname zref@titleref@stripperperiod#1\endcsname
1834 }%
1835 \define@key{ZREF@TR}{cleanup}{%
1836 \zref@titleref@cleanup{#1}%
1837 }%
1838 \define@key{ZREF@TR}{title}{%
1839 \def\zref@titleref@current{#1}%
1840 }%
1841 \ZREF@IfDefinable\ztitlerefsetup\def{%
1842 {\kvsetkeys{ZREF@TR}}%
1843 }%

```

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

```

1844 \ZREF@ifDefinable\ztitleref\def{%
1845   {\zref@wrapper@babel\ZREF@titleref}}%
1846 }%
1847 \def\ZREF@titleref#1{%
1848   \begingroup
1849     \zref@refused{#1}%
1850     \let\label\ltx@gobble
1851     \zref@extract{#1}{title}%
1852   \endgroup
1853 }%

```

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.

```

1854 \AtBeginDocument{%
1855   \ZREF@patch{@caption}{%
1856     \long\def\@caption#1[#2]{%
1857       \zref@titleref@setcurrent{#2}%
1858       \ZREF@org@@caption{#1}[{#2}]}%
1859   }%
1860 }%

```

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

```

1861 \ZREF@patch{@part}{%
1862   \def\@part[#1]{%
1863     \zref@titleref@setcurrent{#1}%
1864     \ZREF@org@@part[{#1}]}%
1865   }%
1866 }%
1867 \ZREF@patch{@chapter}{%
1868   \def\@chapter[#1]{%
1869     \zref@titleref@setcurrent{#1}%
1870     \ZREF@org@@chapter[{#1}]}%
1871   }%
1872 }%
1873 \ZREF@patch{@sect}{%
1874   \def\@sect#1#2#3#4#5#6[#7]{%
1875     \zref@titleref@setcurrent{#7}%
1876     \ZREF@org@@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]}%
1877   }%
1878 }%

```

The star versions of the section commands.

```

1879 \ZREF@patch{@spart}{%
1880   \def\@spart#1{%
1881     \zref@titleref@setcurrent{#1}%
1882     \ZREF@org@@spart{#1}%
1883   }%
1884 }%
1885 \ZREF@patch{@schapter}{%
1886   \def\@schapter#1{%
1887     \zref@titleref@setcurrent{#1}%
1888     \ZREF@org@@schapter{#1}%
1889   }%

```

```

1890 }%
1891 \ZREF@patch{@ssect}{%
1892   \def\ssect#1#2#3#4#5{%
1893     \zref@titleref@setcurrent{#5}%
1894     \ZREF@org@@ssect{#1}{#2}{#3}{#4}{#5}%
1895   }%
1896 }%

```

6.15.4 Environment description

```

1897 \ZREF@patch{descriptionlabel}{%
1898   \def\descriptionlabel#1{%
1899     \zref@titleref@setcurrent{#1}%
1900     \ZREF@org@descriptionlabel{#1}%
1901   }%
1902 }%

```

6.15.5 Class memoir

```

1903 \@ifclassloaded{memoir}{%
1904   \ltx@ifundefined{ifheadnameref}{}{%
1905     \def\chapter[#1]#2{%
1906       \ltx@ifundefined{ch@pt@c}{%
1907         \zref@titleref@setcurrent{#1}%
1908       }{%
1909         \ifx\ch@pt@c\ltx@empty
1910           \zref@titleref@setcurrent{#2}%
1911         \else
1912           \def\NR@temp{#1}%
1913           \ifx\NR@temp\ltx@empty
1914             \expandafter\zref@titleref@setcurrent
1915             \expandafter{\ch@pt@c}%
1916           \else
1917             \ifheadnameref
1918               \zref@titleref@setcurrent{#1}%
1919             \else
1920               \expandafter\zref@titleref@setcurrent
1921               \expandafter{\ch@pt@c}%
1922             \fi
1923           \fi
1924         \fi
1925       }%
1926       \ZREF@org@@chapter[{#1}]{#2}%
1927     }%
1928     \ZREF@patch{M@ssect}{%
1929       \def\M@ssect#1#2#3#4#5#6[#7][#8]{%
1930         \ifheadnameref
1931           \zref@titleref@setcurrent{#8}%
1932         \else
1933           \zref@titleref@setcurrent{#7}%
1934         \fi
1935         \ZREF@org@M@ssect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}][{#8}]%
1936       }%
1937     }%
1938   }%
1939 }{}%

```

6.15.6 Class beamer

```

1940 \@ifclassloaded{beamer}{%

```



```

1941 \ZREF@patch{beamer@section}{%
1942 \long\def\beamer@section[#1]{%
1943 \zref@titleref@setcurrent{#1}%
1944 \ZREF@org@beamer@section[{#1}]%
1945 }%
1946 }%
1947 \ZREF@patch{beamer@subsection}{%
1948 \long\def\beamer@subsection[#1]{%
1949 \zref@titleref@setcurrent{#1}%
1950 \ZREF@org@beamer@subsection[{#1}]%
1951 }%
1952 }%
1953 \ZREF@patch{beamer@subsubsection}{%
1954 \long\def\beamer@subsubsection[#1]{%
1955 \zref@titleref@setcurrent{#1}%
1956 \ZREF@org@beamer@subsubsection[{#1}]%
1957 }%
1958 }%
1959 }{}%

```

6.15.7 Package titlesec

```

1960 \@ifpackageloaded{titlesec}{%
1961 \ZREF@patch{ttl@sect@i}{%
1962 \def\ttl@sect@i#1#2[#3]#4{%
1963 \zref@titleref@setcurrent{#4}%
1964 \ZREF@org@ttl@sect@i{#1}{#2}[{#3}]{#4}%
1965 }%
1966 }%
1967 \ZREF@patch{ttl@straight@i}{%
1968 \def\ttl@straight@i#1[#2]#3{%
1969 \def\ZREF@temp{#2}%
1970 \ifx\ZREF@temp\ltx@empty
1971 \zref@titleref@setcurrent{#3}%
1972 \else
1973 \zref@titleref@setcurrent{#2}%
1974 \fi
1975 \ZREF@org@ttl@straight@i{#1}[{#2}]{#3}%
1976 }%
1977 }%
1978 }{}%

```

6.15.8 Package longtable

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

```

1979 \@ifpackageloaded{longtable}{%
1980 \ZREF@patch{LT@c@ption}{%
1981 \def\LT@c@ption#1[#2]#3{%
1982 \ZREF@org@LT@c@ption{#1}[{#2}]{#3}%
1983 \zref@titleref@setcurrent{#2}%
1984 }%
1985 }%
1986 }{}%

```

6.15.9 Package listings

Package listings: support for its caption.

```

1987 \@ifpackageloaded{listings}{%

```

```

1988 \ZREF@patch{lst@MakeCaption}{%
1989 \def\lst@MakeCaption{%
1990 \ifx\lst@label\ltx@empty
1991 \else
1992 \expandafter\zref@titleref@setcurrent\expandafter{%
1993 \lst@caption
1994 }%
1995 \fi
1996 \ZREF@org@lst@MakeCaption
1997 }%
1998 }%
1999 }{}%

```

6.15.10 Theorems

```

2000 \ZREF@patch{@opargbegintheorem}{%
2001 \def\@opargbegintheorem#1#2#3{%
2002 \zref@titleref@setcurrent{#3}%
2003 \ZREF@org@@opargbegintheorem{#1}{#2}{#3}%
2004 }%
2005 }%
2006 \@ifpackageloaded{amsthm}{%
2007 \begingroup
2008 \edef\x{macro:\string#1\string#2[\string#3]}%
2009 \@onelevel@sanitize\x
2010 \def\y#1->#2\@nil{#1}%
2011 \edef\z{\expandafter\y\meaning\@begintheorem->\@nil}%
2012 \@onelevel@sanitize\z
2013 \expandafter\endgroup
2014 \ifx\x\z
2015 \ZREF@patch{@begintheorem}{%
2016 \def\@begintheorem#1#2[#3]{%
2017 \zref@titleref@setcurrent{#3}%
2018 \ZREF@org@@begintheorem{#1}{#2}[{#3}]}%
2019 }%
2020 }%
2021 \fi
2022 }{}%
2023 }
2024 /<titleref>

```

6.16 Module xr

```

2025 <*xr>
2026 \NeedsTeXFormat{LaTeX2e}
2027 \ProvidesPackage{zref-xr}%
2028 [2020-03-03 v2.29 Module xr for zref (HO)]%
2029 \RequirePackage{zref-base}[2019/11/29]
2030 \ifx\ZREF@base@ok Y%
2031 \else
2032 \expandafter\endinput
2033 \fi
2034 \RequirePackage{keyval}
2035 \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.

```

2036 \zref@newprop{url}{}%
2037 \zref@newprop{urluse}{}%
2038 \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 specied, then assume processed file with a guessed extension.
    Use the setting of hyperref if available.
2039 \providecommand*\zref@xr@ext{%
2040   \ltx@ifundefined{XR@ext}{pdf}{\XR@ext}%
2041 }%

\ifZREF@xr@zreflabel The use of the star form of \zexternaldocument is remembered in the switch
\ifZREF@xr@zreflabel.
2042 \newif\ifZREF@xr@zreflabel

2043 \SetupKeyvalOptions{%
2044   family=ZREF@XR,%
2045   prefix=ZREF@xr@%
2046 }
2047 \DeclareBoolOption[true]{tozreflabel}
2048 \DeclareBoolOption[false]{toltxlabel}
2049 \DeclareBoolOption{verbose}
2050 \define@key{ZREF@XR}{ext}{%
2051   \def\zref@xr@{#1}%
2052 }
2053 \DeclareBoolOption[false]{urluse}

\zxrsetup
2054 \newcommand*\zxrsetup{%
2055   \kvsetkeys{ZREF@XR}%
2056 }%

\ZREF@xr@URL
2057 \newcount\ZREF@xr@URL
2058 \ZREF@xr@URL=\ltx@zero

\ZREF@xr@AddURL
2059 \def\ZREF@xr@AddURL#1{%
2060   \begingroup
2061     \def\ZREF@temp{#1}%
2062     \count@=\ltx@one
2063     \ZREF@xr@@AddUrl
2064   \endgroup
2065 }

\ZREF@xr@@AddUrl
2066 \def\ZREF@xr@@AddUrl{%
2067   \ifnum\count@>\ZREF@xr@URL
2068     \global\advance\ZREF@xr@URL by\ltx@one
2069     \xdef\ZREF@xr@theURL{\romannumeral\ZREF@xr@URL}%
2070     \global\expandafter\let
2071     \csname Z@U@\ZREF@xr@theURL\endcsname\ZREF@temp
2072     \@PackageInfo{zref-xr}{%
2073       \ltx@backslashchar Z@U@\ZREF@xr@theURL:\MessageBreak
2074       \ZREF@temp\MessageBreak
2075     }%

```

```

2076 \else
2077   \expandafter
2078   \ifx\csname Z@U@\romannumeral\count@\endcsname\ZREF@temp
2079   \xdef\ZREF@xr@theURL{\romannumeral\count@}%
2080 \else
2081   \expandafter\expandafter\expandafter\ZREF@xr@@AddUrl
2082   \fi
2083 \fi
2084 }

```

`\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).

```

2085 \ZREF@IfDefinable\zexternaldocument\def{%
2086   {%
2087     \ZREF@NewPropAnchor
2088     \ZREF@NewPropTitle
2089     \begingroup
2090     \csname @safe@actives@true\endcsname
2091     \makeatletter
2092     \@ifstar{%
2093       \ZREF@xr@zreflabelfalse
2094       \@testopt\ZREF@xr@externaldocument{}%
2095     }{%
2096       \ZREF@xr@zreflabeltrue
2097       \@testopt\ZREF@xr@externaldocument{}%
2098     }%
2099   }%
2100 }%

```

If the `\include` featurer 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.

```

2101 \def\ZREF@xr@externaldocument[#1]#2{%
2102   \def\ZREF@xr@prefix{#1}%
2103   \let\ZREF@xr@filelist\ltx@empty
2104   \edef\ZREF@xr@externalfile{#2}%
2105   \edef\ZREF@xr@file{\ZREF@xr@externalfile.aux}%
2106   \filename@parse{#2}%
2107   \@testopt\ZREF@xr@graburl{#2.\zref@xr@ext}%
2108 }%
2109 \def\ZREF@xr@graburl[#1]{%
2110   \edef\ZREF@xr@url{#1}%
2111   \ifZREF@xr@urluse
2112     \expandafter\ZREF@xr@AddURL\expandafter{\ZREF@xr@url}%
2113     \expandafter\def\expandafter\ZREF@xr@url
2114     \expandafter{\csname Z@U@\ZREF@xr@theURL\endcsname}%
2115   \fi
2116   \ZREF@xr@checkfile
2117 \endgroup
2118 }%

```

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

```

2119 \def\ZREF@xr@checkfile{%
2120   \openin\@inputcheck\ZREF@xr@file\relax
2121   \ifeof\@inputcheck
2122     \PackageWarning{zref-xr}{%
2123       File '\ZREF@xr@file' not found or empty,\MessageBreak
2124       labels not imported%
2125     }%
2126   \else
2127     \PackageInfo{zref-xr}{%
2128       Label \ifZREF@xr@zreflabel (zref) \fi
2129       import from '\ZREF@xr@file'%
2130     }%
2131     \def\ZREF@xr@found{0}%
2132     \def\ZREF@xr@ignored@empty{0}%
2133     \def\ZREF@xr@ignored@zref{0}%
2134     \def\ZREF@xr@ignored@ltx{0}%
2135     \ZREF@xr@processfile
2136     \closein\@inputcheck
2137     \begingroup
2138       \let\on@line\ltx@empty
2139       \PackageInfo{zref-xr}{%
2140         Statistics for '\ZREF@xr@file':\MessageBreak
2141         \ZREF@xr@found\space
2142         \ifZREF@xr@zreflabel zref\else LaTeX\fi\space
2143         label(s) found%
2144         \ifnum\ZREF@xr@ignored@empty>0 %
2145           ,\MessageBreak
2146           \ZREF@xr@ignored@empty\space empty label(s) ignored%
2147         \fi
2148         \ifnum\ZREF@xr@ignored@zref>0 %
2149           ,\MessageBreak
2150           \ZREF@xr@ignored@zref\space
2151           duplicated zref label(s) ignored%
2152         \fi
2153         \ifnum\ZREF@xr@ignored@ltx>0 %
2154           ,\MessageBreak
2155           \ZREF@xr@ignored@ltx\space
2156           duplicated latex label(s) ignored%
2157         \fi
2158       }%
2159     \endgroup
2160   \fi
2161   \ifx\ZREF@xr@filelist\ltx@empty
2162   \else
2163     \edef\ZREF@xr@file{%
2164       \expandafter\ltx@car\ZREF@xr@filelist\@nil
2165     }%
2166     \edef\ZREF@xr@filelist{%
2167       \expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil
2168     }%
2169     \expandafter\ZREF@xr@checkfile
2170   \fi
2171 }%

```

\ZREF@xr@processfile

```

2172 \def\ZREF@xr@processfile{%
2173   \read\@inputcheck to\ZREF@xr@line
2174   \expandafter\ZREF@xr@processline\ZREF@xr@line..\ZREF@nil

```

```

2175 \ifeof\@inputcheck
2176 \else
2177 \expandafter\ZREF@xr@processfile
2178 \fi
2179 }%

\ZREF@xr@processline The most work must be done for analyzing the arguments of \newlabel.
2180 \long\def\ZREF@xr@processline#1#2#3\ZREF@nil{%
2181 \def\x{#1}%
2182 \toks@{#2}%
2183 \ifZREF@xr@zreflabel
2184 \ifx\x\ZREF@xr@zref@newlabel
2185 \expandafter
2186 \ZREF@xr@process@zreflabel\ZREF@xr@line...\ZREF@nil
2187 \fi
2188 \else
2189 \ifx\x\ZREF@xr@newlabel
2190 \expandafter
2191 \ZREF@xr@process@label\ZREF@xr@line... [] \ZREF@nil
2192 \fi
2193 \fi
2194 \ifx\x\ZREF@xr@@input
2195 \edef\ZREF@xr@filelist{%
2196 \etex@unexpanded\expandafter{\ZREF@xr@filelist}%
2197 {\filename@area\the\toks@}%
2198 }%
2199 \fi
2200 }%
2201 \def\ZREF@xr@process@zreflabel\zref@newlabel#1#2#3\ZREF@nil{%
2202 \edef\ZREF@xr@refname{Z@R@ZREF@xr@prefix#1}%
2203 \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2204 \def\x{#2}%
2205 \edef\ZREF@xr@tempname{$temp$}%
2206 \edef\ZREF@xr@temprefname{Z@R@ZREF@xr@tempname}%
2207 \let\ZREF@xr@list\x
2208 \ifx\ZREF@xr@list\ltx@empty
2209 \PackageWarningNoLine{zref-xr}{%
2210 Label ‘#1’ without properties ignored\MessageBreak
2211 in file ‘\ZREF@xr@file’%
2212 }%
2213 \edef\ZREF@xr@ignored@empty{%
2214 \the\numexpr\ZREF@xr@ignored@empty+1\relax
2215 }%
2216 \else
2217 \expandafter\ZREF@xr@checklist\x\ZREF@nil
2218 \expandafter\let\csname\ZREF@xr@temprefname\endcsname\x
2219 \expandafter\ltx@LocalAppendToMacro
2220 \csname\ZREF@xr@temprefname\expandafter\endcsname
2221 \expandafter{%
2222 \expandafter\externaldocument\expandafter{%
2223 \ZREF@xr@externalfile
2224 }%
2225 }%
2226 \ZREF@xr@urlcheck\ZREF@xr@tempname
2227 \ifZREF@xr@tozreflabel
2228 \ifundefined{\ZREF@xr@refname}{%
2229 \ifZREF@xr@verbose
2230 \PackageInfo{zref-xr}{%

```

```

2231         Import to zref label '\ZREF@xr@tempname#1'%
2232     }%
2233     \fi
2234     \global\expandafter
2235     \let\csname\ZREF@xr@refname\expandafter\endcsname
2236     \csname\ZREF@xr@temprefname\endcsname
2237 }{%
2238     \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2239 }%
2240 \fi
2241 \ifZREF@xr@toltxlabel
2242     \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2243 \fi
2244 \fi
2245 }%
2246 \def\ZREF@xr@process@label\newlabel#1#2#3[#4]#5\ZREF@nil{%
2247     \def\ZREF@xr@refname{Z@R@{\ZREF@xr@prefix#1}%
2248     \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2249     \def\x{#2}%
2250     \edef\ZREF@xr@tempname{$temp$}%
2251     \edef\ZREF@xr@temprefname{Z@R@{\ZREF@xr@tempname}%
2252     \expandafter\ZREF@xr@scanparams
2253         \csname\ZREF@xr@temprefname\expandafter\endcsname
2254         \x{}{}{}{}{}\ZREF@nil
2255     \ifx\\#4\\%
2256     \else
2257         % ntheorem knows an optional argument at the end of \newlabel
2258         \ZREF@NewPropTheotype
2259         \expandafter\ltx@LocalAppendToMacro
2260         \csname\ZREF@xr@temprefname\endcsname{\theotype{#4}}%
2261     \fi
2262     \expandafter\ltx@LocalAppendToMacro
2263     \csname\ZREF@xr@temprefname\expandafter\endcsname\expandafter{%
2264     \expandafter\externaldocument\expandafter{%
2265     \ZREF@xr@externalfile
2266     }%
2267 }%
2268 \ZREF@xr@urlcheck\ZREF@xr@tempname
2269 \ifZREF@xr@tozreflabel
2270     \ifundefined{\ZREF@xr@refname}{%
2271     \ifZREF@xr@verbose
2272         \PackageInfo{zref-xr}{%
2273         Import to zref label '\ZREF@xr@prefix#1'%
2274         }%
2275     \fi
2276     \global\expandafter
2277     \let\csname\ZREF@xr@refname\expandafter\endcsname
2278     \csname\ZREF@xr@temprefname\endcsname
2279 }{%
2280     \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2281 }%
2282 \fi
2283 \ifZREF@xr@toltxlabel
2284     \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2285 \fi
2286 }
2287 \def\ZREF@xr@zref@newlabel{\zref@newlabel}%
2288 \def\ZREF@xr@newlabel{\newlabel}%

```

```

2289 \def\ZREF@xr@input{\@input}%
2290 \def\ZREF@xr@relax{\relax}%

```

\ZREF@xr@tolabel

```

2291 \def\ZREF@xr@tolabel#1#2{%
2292   \ifZREF@xr@verbose
2293     \PackageInfo{zref-xr}{%
2294       Import to LaTeX label ‘#2’%
2295     }%
2296   \fi
2297   \zref@wrapper@unexpanded{%
2298     \expandafter\xdef\csname r@#2\endcsname{%
2299       {%
2300         \ltx@ifundefined{M@TitleReference}{%
2301           \ltx@ifundefined{TR@TitleReference}{%
2302             \zref@extractdefault{#1}{default}{}}%
2303           {%
2304             \noexpand\TR@TitleReference
2305             {\zref@extractdefault{#1}{default}{}}}%
2306             {\zref@extractdefault{#1}{title}{}}}%
2307           }%
2308         }{%
2309           \noexpand\M@TitleReference
2310           {\zref@extractdefault{#1}{default}{}}}%
2311           {\zref@extractdefault{#1}{title}{}}}%
2312         }%
2313       }%
2314       {\zref@extractdefault{#1}{page}{}}}%
2315     \ltx@ifpackageloaded{nameref}{%
2316       {\zref@extractdefault{#1}{title}{}}}%
2317       {\zref@extractdefault{#1}{anchor}{}}}%
2318     \zref@ifrefcontainsprop{#1}{urluse}{%
2319       {\zref@extractdefault{#1}{urluse}{}}}%
2320     }{%
2321       {\zref@extractdefault{#1}{url}{}}}%
2322     }%
2323   }{}%
2324 }%
2325 }%
2326 }

```

\ZREF@xr@zref@ignorewarning

```

2327 \def\ZREF@xr@zref@ignorewarning#1{%
2328   \PackageWarningNoLine{zref-xr}{%
2329     Zref label ‘#1’ is already in use\MessageBreak
2330     in file ‘\ZREF@xr@file’%
2331   }%
2332   \edef\ZREF@xr@ignored@zref{%
2333     \the\numexpr\ZREF@xr@ignored@zref+1%
2334   }%
2335 }%

```

\ZREF@xr@ltx@ignorewarning

```

2336 \def\ZREF@xr@ltx@ignorewarning#1{%
2337   \PackageWarningNoLine{zref-xr}{%
2338     LaTeX label ‘#1’ is already in use\MessageBreak
2339     in file ‘\ZREF@xr@file’%
2340   }%

```



```

2341 \edef\ZREF@xr@ignored@ltx{%
2342 \the\numexpr\ZREF@xr@ignored@ltx+1%
2343 }%
2344 }%

```

\ZREF@xr@checklist

```

2345 \def\ZREF@xr@checklist#1#2#3\ZREF@nil{%
2346 \ifx\@undefined#1\relax
2347 \expandafter\ZREF@xr@checkkey\string#1\@nil
2348 \fi
2349 \ifx\\#3\\%
2350 \else
2351 \ltx@ReturnAfterFi{%
2352 \ZREF@xr@checklist#3\ZREF@nil
2353 }%
2354 \fi
2355 }%
2356 \def\ZREF@xr@checkkey#1#2\@nil{%
2357 \zref@ifpropundefined{#2}{%
2358 \zref@newprop{#2}{}%
2359 }{}%
2360 }%

```

\ZREF@xr@scanparams

```

2361 \def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{%
2362 \let#1\ltx@empty
2363 \ZREF@foundfalse
2364 \ZREF@xr@scantitleref#1#2\TR@TitleReference{}{}\ZREF@nil
2365 \ifZREF@found
2366 \else
2367 \ltx@LocalAppendToMacro#1{\default{#2}}%
2368 \fi
2369 % page
2370 \ltx@LocalAppendToMacro#1{\page{#3}}%
2371 % nameref title
2372 \ifZREF@found
2373 \else
2374 \ifx\\#4\\%
2375 \else
2376 \def\ZREF@xr@temp{#4}%
2377 \ifx\ZREF@xr@temp\ZREF@xr@relax
2378 \else
2379 \ltx@LocalAppendToMacro#1{\title{#4}}%
2380 \fi
2381 \fi
2382 \fi
2383 % anchor
2384 \ifx\\#5\\%
2385 \else
2386 \ltx@LocalAppendToMacro#1{\anchor{#5}}%
2387 \fi
2388 \ifx\\#6\\%
2389 \else
2390 \ifZREF@xr@urluse
2391 \ZREF@xr@AddURL{#6}%
2392 \expandafter\ltx@LocalAppendToMacro\expandafter#1%
2393 \expandafter{%
2394 \expandafter\urluse\expandafter{%

```

```

2395         \csname Z@U@\ZREF@xr@theURL\endcsname
2396     }%
2397 }%
2398 \else
2399     \ltx@LocalAppendToMacro#1{\url{#6}}%
2400 \fi
2401 \fi
2402 }%

```

\ZREF@xr@scantitleref

```

2403 \def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{%
2404     \ifx\#5\%
2405     \else
2406         \ltx@LocalAppendToMacro#1{%
2407             \default{#3}%
2408             \title{#4}%
2409         }%
2410         \ZREF@foundtrue
2411     \fi
2412 }%

```

\ZREF@xr@urlcheck

```

2413 \def\ZREF@xr@urlcheck#1{%
2414     \zref@ifrefcontainsprop{#1}{anchor}{%
2415         \zref@ifrefcontainsprop{#1}{url}{%
2416             }{%
2417                 \expandafter
2418                 \ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
2419                 \expandafter{%
2420                     \csname url\ifZREF@xr@urluse use\fi
2421                     \expandafter\endcsname\expandafter{\ZREF@xr@url}%
2422                 }%
2423             }%
2424         }{%
2425         }%
2426     }%
2427 </xr>

```

6.17 Module hyperref

UNFINISHED :-(

```

2428 (*hyperref)
2429 \NeedsTeXFormat{LaTeX2e}
2430 \ProvidesPackage{zref-hyperref}%
2431     [2020-03-03 v2.29 Module hyperref for zref (H0)]%
2432 \RequirePackage{zref-base}[2019/11/29]
2433 \ifx\ZREF@base@ok Y%
2434 \else
2435     \expandafter\endinput
2436 \fi
2437 \ZREF@NewPropAnchor
2438 \zref@addprop\ZREF@mainlist{anchor}%
2439 </hyperref>

```

6.18 Module `savepos`

Module `savepos` provides an interface for pdfTeX's `\pdfsavepos`, see the manual for pdfTeX.

6.18.1 Identification

```
2440 (*savepos)
2441 \NeedsTeXFormat{LaTeX2e}
2442 \ProvidesPackage{zref-savepos}%
2443   [2020-03-03 v2.29 Module savepos for zref (H0)]%
2444 \RequirePackage{zref-base}[2019/11/29]
2445 \ifx\ZREF@base@ok Y%
2446 \else
2447   \expandafter\endinput
2448 \fi
```

6.18.2 Availability

First we check, whether the feature is available.

```
2449 \ifx\directlua\@undefined
2450 \ltx@ifundefined{pdfsavepos}{%
2451   \PackageError\ZREF@name{%
2452     \string\pdfsavepos\space is not supported.\MessageBreak
2453     It is provided by pdfTeX (1.40) or XeTeX%
2454   }\ZREF@UpdatePdfTeX
2455   \endinput
2456 }{}%
2457 \fi
```

In PDF mode we are done. However support for DVI mode was added later in version 1.40.0. In earlier versions `\pdfsavepos` is defined, but its execution raises an error. Note that XeTeX also provides `\pdfsavepos`.

```
2458 \ifpdf
2459 \else
2460   \ltx@ifundefined{pdftexversion}{%
2461     }{}%
2462     \ifnum\pdftexversion<140 %
2463       \PackageError\ZREF@name{%
2464         \string\pdfsavepos\space is not supported in DVI mode%
2465         \MessageBreak
2466         of this pdfTeX version%
2467       }\ZREF@UpdatePdfTeX
2468       \expandafter\expandafter\expandafter\endinput
2469     \fi
2470   }%
2471 \fi
```

6.18.3 Setup

```
2472 \zref@newlist{savepos}
2473 \ifx\directlua\@undefined
2474   \zref@newprop*{posx}[0]{\the\pdflastxpos}
2475   \zref@newprop*{posy}[0]{\the\pdflastypos}
2476 \else
2477   \zref@newprop*{posx}[0]{\the\lastxpos}
2478   \zref@newprop*{posy}[0]{\the\lastypos}
2479 \fi
2480 \zref@addprops{savepos}{posx,posy}
```

6.18.4 User macros

`\zref@savepos`

```

2481 \ifx\directlua\@undefined
2482   \def\zref@savepos{%
2483     \if@files
2484       \pdfsavepos
2485     \fi
2486   }
2487 \else
2488   \def\zref@savepos{%
2489     \if@files
2490       \savepos
2491     \fi
2492   }
2493 \fi

```

`\ZREF@zsavepos`

```

2494 \def\ZREF@zsavepos#1#2#3{%
2495   \@bsphack
2496   \if@files
2497     \zref@savepos
2498     #1{#3}{#2}%
2499     \ltx@ifundefined{TeXeTstate}{%
2500     }{%
2501       \ifnum\TeXeTstate=\ltx@zero
2502       \else
2503         \zref@savepos
2504       \fi
2505     }%
2506   \fi
2507   \@esphack
2508 }

```

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

```

2509 \ZREF@ifDefinable\zsavepos\def{%
2510   {%
2511     \ZREF@zsavepos\zref@labelbylist{savepos}%
2512   }%
2513 }

```

`\zsaveposx`

```

2514 \ZREF@ifDefinable\zsaveposx\def{%
2515   {%
2516     \ZREF@zsavepos\zref@labelbyprops{posx}%
2517   }%
2518 }

```

`\zsaveposy`

```

2519 \ZREF@ifDefinable\zsaveposy\def{%
2520   {%
2521     \ZREF@zsavepos\zref@labelbyprops{posy}%
2522   }%
2523 }

```

`\zposx` 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.

```

2524 \newcommand*{\zposx}[1]{%
2525   \zref@extract{#1}{posx}%
2526 }%
2527 \newcommand*{\zposy}[1]{%
2528   \zref@extract{#1}{posy}%
2529 }%

```

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 marked used positions by `\zrefused` in order to notify L^AT_EX about undefined references.

`\ZREF@savepos@ok`

```

2530 \let\ZREF@savepos@ok=Y
2531 \savepos

```

6.19 Module `abspos`

6.19.1 Identification

```

2532 (*abspos)
2533 \NeedsTeXFormat{LaTeX2e}
2534 \ProvidesPackage{zref-abspos}%
2535   [2020-03-03 v2.29 Module abspos for zref (H0)]%
2536 \RequirePackage{zref-base}[2019/11/29]
2537 \ifx\ZREF@base@ok Y%
2538 \else
2539   \expandafter\endinput
2540 \fi

2541 \RequirePackage{zref-savepos}[2019/11/29]
2542 \ifx\ZREF@savepos@ok Y%
2543 \else
2544   \expandafter\endinput
2545 \fi

2546 \RequirePackage{zref-pagelayout}[2019/11/29]
2547 \zref@addprop{savepos}{abspage}

```

`\zref@absposx`

```

2548 \newcommand*{\zref@absposx}[3]{%
2549   \number
2550   \expandafter\zref@absposnumx\expandafter{%
2551     \number\zref@extractdefault{#1}{abspage}{0}%
2552   }{#2}{#3}%
2553   \ltx@space
2554 }

```

`\zref@absposy`

```

2555 \newcommand*{\zref@absposy}[3]{%
2556   \number
2557   \expandafter\zref@absposnumy\expandafter{%
2558     \number\zref@extractdefault{#1}{abspage}{0}%
2559   }{#2}{#3}%
2560   \ltx@space
2561 }

```

\zref@absposnumx

```
2562 \newcommand*{\zref@absposnumx}[3]{%
2563   \number
2564 %   \ifnum#1>\ltx@zero
2565 %     \zref@ifrefundefined{thepage#1}{%
2566 %       0%
2567 %     }{%
2568       \numexpr\ZREF@absposnum{thepage#1}{#2}{x}{#3}\relax
2569 %     }%
2570 %   \else
2571 %     0%
2572 %   \fi
2573 }
```

\zref@absposnumy

```
2574 \newcommand*{\zref@absposnumy}[3]{%
2575   \number
2576 %   \ifnum#1>\ltx@zero
2577 %     \zref@ifrefundefined{thepage#1}{%
2578 %       0%
2579 %     }{%
2580       \numexpr\ZREF@absposnum{thepage#1}{#2}{y}{#3}\relax
2581 %     }%
2582 %   \else
2583 %     0%
2584 %   \fi
2585 }
```

\ZREF@absposnum

```
2586 \def\ZREF@absposnum#1#2#3#4{%
2587   \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
2588     0%
2589   }{%
2590     \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
2591   }%
2592 }
```

\zref@def@absposx

```
2593 \ZREF@Robust\def\zref@def@absposx#1{%
2594   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposx}%
2595 }
```

\zref@def@absposy

```
2596 \ZREF@Robust\def\zref@def@absposy#1{%
2597   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
2598 }
```

\zref@def@absposnumx

```
2599 \ZREF@Robust\def\zref@def@absposnumx#1{%
2600   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumx}%
2601 }
```

\zref@def@absposnumy

```
2602 \ZREF@Robust\def\zref@def@absposnumy#1{%
2603   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumy}%
2604 }
```

\ZREF@def@abspos

```
2605 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2606   \edef#1{#2{#3}{#4}{#5}}}%
2607 }
```

\zref@absposused

```
2608 \ZREF@Robust\def\zref@absposused{%
2609   \zref@wrapper@babel\ZREF@abspos@used
2610 }
```

\ZREF@abspos@used

```
2611 \def\ZREF@abspos@used#1{%
2612   \zref@refused{#1}%
2613   \zref@ifrefundefined{#1}{%
2614     }{%
2615     \begingroup
2616       \edef\ZREF@temp{%
2617         \zref@extractdefault{#1}{abspage}{0}}%
2618       }%
2619       \ifnum\ZREF@temp>\ltx@zero
2620         \zref@refused{thepage\ZREF@temp}%
2621       \else
2622         \@PackageError{zref-abspos}{%
2623           \string\zref@pos@label@used\ltx@space
2624           needs property 'abspage'\MessageBreak
2625           in label '#1'%
2626         }\@ehc
2627       \fi
2628     \endgroup
2629   }%
2630 }
```

\zref@absposnumused

```
2631 \newcommand*{\zref@absposnumused}[1]{%
2632   \ifnum#1>\ltx@zero
2633     \zref@refused{thepage\number#1}%
2634   \else
2635     \@PackageError{zref-abspos}{%
2636       Invalid absolute page number (#1)\MessageBreak
2637       for \string\zref@pos@num@used.\MessageBreak
2638       A positive integer is expected%
2639     }\@ehc
2640   \fi
2641 }
```

\zref@ifabsposundefined

```
2642 \def\zref@ifabsposundefined#1{%
2643   \zref@ifrefundefined{#1}\ltx@firsttwo{%
2644     \expandafter\zref@ifabsposnumundefined\expandafter{%
2645       \number\zref@extractdefault{#1}{abspage}{0}}%
2646     }%
2647   }%
2648 }
```

\zref@ifabsposnumundefined

```
2649 \def\zref@ifabsposnumundefined#1{%
2650   \ifnum\ZREF@number{#1}>\ltx@zero
```

```

2651 \zref@ifrefundefined{thepage#1}%
2652 \ltx@iffirstoftwo\ltx@secondoftwo
2653 \else
2654 \expandafter\ltx@iffirstoftwo
2655 \fi
2656 }

```

6.19.2 Media

`\ZREF@abspos@media@width`

```

2657 \edef\ZREF@abspos@media@width{%
2658 \ltx@ifundefined{pdfpagewidth}{%
2659 \ltx@ifundefined{mediawidth}{%
2660 \ltx@ifundefined{stockwidth}{%
2661 paperwidth%
2662 }{%
2663 stockwidth%
2664 }%
2665 }{%
2666 mediawidth%
2667 }%
2668 }{%
2669 pdfpagewidth%
2670 }%
2671 }
2672 \ifluatex
2673 \def\ZREF@abspos@media@width{pdfpagewidth}%
2674 \fi

```

`\ZREF@abspos@media@height`

```

2675 \edef\ZREF@abspos@media@height{%
2676 \ltx@ifundefined{pdfpageheight}{%
2677 \ltx@ifundefined{mediaheight}{%
2678 \ltx@ifundefined{stockheight}{%
2679 paperheight%
2680 }{%
2681 stockheight%
2682 }%
2683 }{%
2684 mediaheight%
2685 }%
2686 }{%
2687 \noexpand\ifcase\pdfpageheight
2688 \ltx@ifundefined{stockheight}{%
2689 paperheight%
2690 }{%
2691 stockheight%
2692 }%
2693 \noexpand\else
2694 pdfpageheight%
2695 \noexpand\fi
2696 }%
2697 }
2698 \ifluatex
2699 \edef\ZREF@abspos@media@height{%
2700 \noexpand\ifcase\pageheight
2701 \ltx@ifundefined{stockheight}{%
2702 paperheight%

```



```

2703      }{%
2704      stockheight%
2705      }%
2706      \noexpand\else
2707      pdfpageheight%
2708      \noexpand\fi}%
2709 \fi

```

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

```

2710 \def\ZREF@abspos@media@x@left#1{%
2711   0%
2712 }

```

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

```

2713 \def\ZREF@abspos@media@x@right#1{%
2714   \zref@extract{#1}\ZREF@abspos@media@width
2715 }

```

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

```

2716 \def\ZREF@abspos@media@x@center#1{%
2717   \ZREF@abspos@media@x@left{#1}%
2718   +\zref@extract{#1}\ZREF@abspos@media@width/2%
2719 }

```

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

```

2720 \def\ZREF@abspos@media@y@top#1{%
2721   \zref@extract{#1}\ZREF@abspos@media@height
2722 }

```

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

```

2723 \def\ZREF@abspos@media@y@bottom#1{%
2724   0%
2725 }

```

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

```

2726 \def\ZREF@abspos@media@y@center#1{%
2727   \zref@extract{#1}\ZREF@abspos@media@height/2%
2728 }

```

6.19.3 Paper

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

```

2729 \def\ZREF@abspos@paper@x@left#1{%
2730   0%
2731 }

```

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

```

2732 \def\ZREF@abspos@paper@x@right#1{%
2733   \zref@extract{#1}{paperwidth}%
2734 }

```

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

```

2735 \def\ZREF@abspos@paper@x@center#1{%
2736   \zref@extract{#1}{paperwidth}/2%
2737 }

```

\ZREF@abspos@paper@y@top

```
2738 \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top
```

\ZREF@abspos@paper@y@bottom

```
2739 \def\ZREF@abspos@paper@y@bottom#1{%
2740   \ZREF@abspos@paper@y@top{#1}%
2741   -\zref@extract{#1}{paperheight}%
2742 }
```

\ZREF@abspos@paper@y@center

```
2743 \def\ZREF@abspos@paper@y@center#1{%
2744   \ZREF@abspos@paper@y@top{#1}%
2745   -\zref@extract{#1}{paperheight}/2%
2746 }
```

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

```
2747 \let\ZREF@temp\ltx@two
2748 \ltx@ifundefined{pdfhorigin}{-}{%
2749 %   \ifpdf
2750     \let\ZREF@temp\ltx@zero
2751 %   \fi
2752 }
2753 \ifluatex
2754 % \ifpdf
2755   \let\ZREF@temp\ltx@zero
2756 % \fi
2757 \fi
2758
2759 \ifx\ZREF@temp\ltx@two
2760   \ifnum\mag=1000 %
2761     \let\ZREF@temp\ltx@one
2762   \fi
2763 \fi
2764 \ifcase\ZREF@temp
2765   \def\ZREF@abspos@origin@x#1{%
2766     \zref@extract{#1}{pdfhorigin}%
2767   }%
2768 \or
2769   \def\ZREF@abspos@origin@x#1{%
2770     4736286%
2771   }%
2772 \or
2773   \def\ZREF@abspos@origin@x#1{%
2774     \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2775   }%
2776 \fi
```

\ZREF@abspos@origin@y

```
2777 \let\ZREF@temp\ltx@two
2778 \ltx@ifundefined{pdfvorigin}{-}{%
2779 %   \ifpdf
2780     \let\ZREF@temp\ltx@zero
2781 %   \fi
```

```

2782 }
2783 \ifluatex
2784 % \ifpdf
2785 \let\ZREF@temp\ltx@zero
2786 % \fi
2787 \fi
2788 \ifx\ZREF@temp\ltx@two
2789 \ifnum\mag=1000 %
2790 \let\ZREF@temp\ltx@one
2791 \fi
2792 \fi
2793 \ifcase\ZREF@temp
2794 \def\ZREF@abspos@origin@y#1{%
2795 \zref@extract{#1}{pdfvorigin}%
2796 }%
2797 \or
2798 \def\ZREF@abspos@origin@y#1{%
2799 4736286%
2800 }%
2801 \or
2802 \def\ZREF@abspos@origin@y#1{%
2803 \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2804 }%
2805 \fi

```

6.19.5 Header

\ZREF@abspos@head@x@left

```

2806 \def\ZREF@abspos@head@x@left#1{%
2807 \ZREF@abspos@paper@x@left{#1}%
2808 +\ZREF@abspos@origin@x{#1}%
2809 +\zref@extract{#1}{hoffset}%
2810 +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2811 \zref@extract{#1}{oddsidemargin}%
2812 \else
2813 \zref@extract{#1}{evensidemargin}%
2814 \fi
2815 }

```

\ZREF@abspos@head@x@right

```

2816 \def\ZREF@abspos@head@x@right#1{%
2817 \ZREF@abspos@head@x@left{#1}%
2818 +\zref@extract{#1}{textwidth}%
2819 }

```

\ZREF@abspos@head@x@center

```

2820 \def\ZREF@abspos@head@x@center#1{%
2821 \ZREF@abspos@head@x@left{#1}%
2822 +\zref@extract{#1}{textwidth}/2%
2823 }

```

\ZREF@abspos@head@y@top

```

2824 \def\ZREF@abspos@head@y@top#1{%
2825 \ZREF@abspos@paper@y@top{#1}%
2826 -\ZREF@abspos@origin@y{#1}%
2827 -\zref@extract{#1}{voffset}%
2828 -\zref@extract{#1}{topmargin}%
2829 }

```

`\ZREF@abspos@head@y@bottom`

```
2830 \def\ZREF@abspos@head@y@bottom#1{%
2831   \ZREF@abspos@head@y@top{#1}%
2832   -\zref@extract{#1}{headheight}%
2833 }
```

`\ZREF@abspos@head@y@center`

```
2834 \def\ZREF@abspos@head@y@center#1{%
2835   \ZREF@abspos@head@y@top{#1}%
2836   -\zref@extract{#1}{headheight}/2%
2837 }
```

6.19.6 Body

`\ZREF@abspos@body@x@left`

```
2838 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left
```

`\ZREF@abspos@body@x@right`

```
2839 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right
```

`\ZREF@abspos@body@x@center`

```
2840 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center
```

`\ZREF@abspos@body@y@top`

```
2841 \def\ZREF@abspos@body@y@top#1{%
2842   \ZREF@abspos@head@y@bottom{#1}%
2843   -\zref@extract{#1}{headsep}%
2844 }
```

`\ZREF@abspos@body@y@bottom`

```
2845 \def\ZREF@abspos@body@y@bottom#1{%
2846   \ZREF@abspos@body@y@top{#1}%
2847   -\zref@extract{#1}{textheight}%
2848 }
```

`\ZREF@abspos@body@y@center`

```
2849 \def\ZREF@abspos@body@y@center#1{%
2850   \ZREF@abspos@body@y@top{#1}%
2851   -\zref@extract{#1}{textheight}/2%
2852 }
```

6.19.7 Footer

`\ZREF@abspos@foot@x@left`

```
2853 \let\ZREF@abspos@foot@x@left\ZREF@abspos@head@x@left
```

`\ZREF@abspos@foot@x@right`

```
2854 \let\ZREF@abspos@foot@x@right\ZREF@abspos@head@x@right
```

`\ZREF@abspos@foot@x@center`

```
2855 \let\ZREF@abspos@foot@x@center\ZREF@abspos@head@x@center
```

`\ZREF@abspos@foot@y@bottom`

```
2856 \def\ZREF@abspos@foot@y@bottom#1{%
2857   \ZREF@abspos@body@y@bottom{#1}%
2858   -\zref@extract{#1}{footskip}%
2859 }
```

6.19.8 Marginal notes

```
\ZREF@abspos@marginpar@x@left
2860 \def\ZREF@abspos@marginpar@x@left#1{%
2861   \ifodd\zref@extractdefault{#1}{\pagevalue}{\number\c@page} %
2862   \ZREF@abspos@body@x@right{#1}%
2863   +\zref@extract{#1}{marginparsep}%
2864   \else
2865     \ZREF@abspos@body@x@left{#1}%
2866     -\zref@extract{#1}{marginparsep}%
2867     -\zref@extract{#1}{marginparwidth}%
2868   \fi
2869 }

\ZREF@abspos@marginpar@x@right
2870 \def\ZREF@abspos@marginpar@x@right#1{%
2871   \ZREF@abspos@marginpar@x@left{#1}%
2872   +\zref@extract{#1}{marginparwidth}%
2873 }

\ZREF@abspos@marginpar@x@center
2874 \def\ZREF@abspos@marginpar@x@center#1{%
2875   \ZREF@abspos@marginpar@x@left{#1}%
2876   +\zref@extract{#1}{marginparwidth}/2%
2877 }

\ZREF@abspos@marginpar@y@top
2878 \let\ZREF@abspos@marginpar@y@top\ZREF@abspos@body@y@top

\ZREF@abspos@marginpar@y@bottom
2879 \let\ZREF@abspos@marginpar@y@bottom\ZREF@abspos@body@y@bottom

\ZREF@abspos@marginpar@y@center
2880 \let\ZREF@abspos@marginpar@y@center\ZREF@abspos@body@y@center

6.19.9 Stock paper

\ZREF@abspos@stock@x@left
2881 \let\ZREF@abspos@stock@x@left\ZREF@abspos@paper@x@left

\ZREF@abspos@stock@x@right
2882 \let\ZREF@abspos@stock@x@right\ZREF@abspos@paper@x@right

\ZREF@abspos@stock@x@center
2883 \let\ZREF@abspos@stock@x@center\ZREF@abspos@paper@x@center

\ZREF@abspos@stock@y@top
2884 \let\ZREF@abspos@stock@y@top\ZREF@abspos@paper@y@top

\ZREF@abspos@stock@y@bottom
2885 \let\ZREF@abspos@stock@y@bottom\ZREF@abspos@paper@y@bottom

\ZREF@abspos@stock@y@center
2886 \let\ZREF@abspos@stock@y@center\ZREF@abspos@paper@y@center
2887 </abspos>
```

6.20 Module dotfill

```

2888 (*dotfill)
2889 \NeedsTeXFormat{LaTeX2e}
2890 \ProvidesPackage{zref-dotfill}%
2891   [2020-03-03 v2.29 Module dotfill for zref (HO)]%
2892 \RequirePackage{zref-base}[2019/11/29]
2893 \ifx\ZREF@base@ok Y%
2894 \else
2895   \expandafter\endinput
2896 \fi

```

For measuring the width of `\zdotfill` we use the features provided by module `savepos`.

```

2897 \RequirePackage{zref-savepos}[2019/11/29]

```

For automatically generated label names we use the unique counter of module `base`.

```

2898 \zref@require@unique

```

Configuration is done by the key value interface of package `keyval`.

```

2899 \RequirePackage{keyval}

```

The definitions of the keys follow.

```

2900 \define@key{ZREF@DF}{unit}{%
2901   \def\ZREF@df@unit{#1}%
2902 }
2903 \define@key{ZREF@DF}{min}{%
2904   \def\ZREF@df@min{#1}%
2905 }
2906 \define@key{ZREF@DF}{dot}{%
2907   \def\ZREF@df@dot{#1}%
2908 }

```

Defaults are set, see user interface.

```

2909 \providecommand\ZREF@df@min{2}
2910 \providecommand\ZREF@df@unit{.44em}
2911 \providecommand\ZREF@df@dot{.}

```

`\zdotfillsetup` Configuration of `\zdotfill` is done by `\zdotfillsetup`.

```

2912 \newcommand*{\zdotfillsetup}{\kvsetkeys{ZREF@DF}}

```

`\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.

```

2913 \ZREF@ifDefinable\zdotfill\def{%
2914   {%
2915     \leavevmode
2916     \global\advance\c@zref@unique\ltx@one
2917     \begingroup
2918       \def\ZREF@temp{zref@\number\c@zref@unique}%
2919       \pdfsavepos
2920       \zref@labelbyprops{\thezref@unique L}{posx}%
2921       \setlength{\dimen@}{\ZREF@df@unit}%
2922       \zref@ifrefundefined{\thezref@unique R}{%
2923         \ZREF@dotfill
2924       }{%
2925         \ifnum\numexpr\zposx{\thezref@unique R}%
2926           -\zposx{\thezref@unique L}\relax
2927         <\dimexpr\ZREF@df@min\dimen@\relax
2928         \hfill
2929       }%
2930       \ZREF@dotfill

```

```

2931      \fi
2932    }%
2933    \pdfsavepos
2934    \zref@labelbyprops{\thezref@unique R}{posx}%
2935  \endgroup
2936  \kern\z@
2937  }%
2938 }

```

`\ZREF@dotfill` Help macro that actually sets the dots.

```

2939 \def\ZREF@dotfill{%
2940   \cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill
2941 }

2942 </dotfill>

```

6.21 Module `env`

```

2943 (*env)
2944 \NeedsTeXFormat{LaTeX2e}
2945 \ProvidesPackage{zref-env}%
2946   [2020-03-03 v2.29 Module env for zref (HO)]%
2947 \RequirePackage{zref-base}[2019/11/29]
2948 \ifx\ZREF@base@ok Y%
2949 \else
2950   \expandafter\endinput
2951 \fi

2952 \zref@newprop{envname}[]{\@currentvir}
2953 \zref@newprop{envline}[]{\zref@env@line}

```

`\zref@env@line` Macro `\zref@env@line` extracts the line number from `\@currentvline`.

```

2954 \def\zref@env@line{%
2955   \ifx\@currentvline\ltx@empty
2956   \else
2957     \expandafter
2958     \ZREF@ENV@line\@currentvline\ltx@empty line \ltx@empty\@nil
2959   \fi
2960 }

```

`\ZREF@ENV@line`

```

2961 \def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%
2962 </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)

Bundle. All the packages of the bundle ‘zref’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/zref.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

7.2 Bundle installation

Unpacking. Unpack the `zref.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip zref.tds.zip -d ~/texmf
```

7.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain T_EX:

```
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`.

7.4 Refresh file name databases

If your \TeX distribution (\TeX Live, MiK \TeX , ...) relies on file name databases, you must refresh these. For example, \TeX Live users run `texhash` or `mktextlsr`.

7.5 Some details for the interested

Unpacking with \LaTeX . The `.dtx` chooses its action depending on the format:

plain \TeX : Run `docstrip` and extract the files.

\LaTeX : Generate the documentation.

If you insist on using \LaTeX for `docstrip` (really, `docstrip` does not need \LaTeX), 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 pdf \LaTeX :

```
pdflatex zref.dtx
makeindex -s gind.ist zref.idx
pdflatex zref.dtx
makeindex -s gind.ist zref.idx
pdflatex 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 Arsenau, 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 \TeX .

[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 ε -TeX'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 `getttitlestring`.

[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]

- `\externaldocument` 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@{local}addprop(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 `lualatex` 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.

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	
\@PackageError ..	508, 524, 2622, 2635
\@PackageInfo	2072
\@PackageInfoNoLine	548,
	563, 1336, 1431, 1443, 1511, 1563
\@PackageWarning	691
\@addtoreset	911, 1008
\@auxout	715
\@begintheorem	2011, 2016
\@bsphack	600, 610, 630, 2495
\@caption	1856
\@chapter	1868, 1905
\@currentHref	940
\@currentlabel	935
\@currenvir	2952
\@currentvline	2955, 2958
\@ehc	296,
	306, 491, 514, 526, 1595, 2626, 2639
\@esphack	607, 627, 642, 2507
\@firstofone	1551
\@firstoftwo	1416, 1495
\@ifclassloaded	1903, 1940
\@ifdefinable	242, 289
\@ifnextchar	530, 1722
\@ifpackageloaded	
	1960, 1979, 1987, 2006
\@ifstar	495, 2092
\@ifundefined	192, 909, 1733, 2228, 2270
\@input	2289
\@inputcheck	
	2120, 2121, 2136, 2173, 2175
\@latex@warning	777
\@mainaux	1680
\@namedef	535
\@newl@bel	285
\@nil	1610,
	1612, 1821, 1828, 2010, 2011,
	2164, 2167, 2347, 2356, 2958, 2961
\@onelevel@sanitize	
	422, 440, 505, 533, 2009, 2012
\@opargbegintheorem	2001
\@part	1862
\@schapter	1886
\@secondoftwo	1418, 1497
\@sect	1874
\@spart	1880
\@ssect	1892
\@stpelt	1709, 1710
\@testdef	1318, 1319, 1479, 1480
\@testopt	2094, 2097, 2107
\@tfor	322, 720
\@undefined	1777, 2346, 2449, 2473, 2481
\\	25, 26, 27,
	28, 153, 155, 157, 158, 170, 173,
	2255, 2349, 2374, 2384, 2388, 2404
_	44, 45
A	
\AddLineBeginAux	280
\advance	
	1053, 1383, 1540, 1746, 2068, 2916
\afterassignment	233, 1137, 1141
\AfterLastShipout ..	1050, 1400, 1572
\Alph	7
\anchor	2386
\AtBeginDocument	1029, 1226, 1674, 1854
\AtBeginShipout	1012, 1097
\AtBeginShipoutAddToBox	1098
\AtBeginShipoutBoxDepth	1308
\AtBeginShipoutBoxHeight	1307
\AtBeginShipoutBoxWidth	1306
\AtEndOfPackage	195
\AtVeryEndDocument	1327, 1488
B	
\beamer@section	1942
\beamer@subsection	1948
\beamer@subsubsection	1954
\begin	23, 57, 100, 106, 156, 172
\bfseries	928
C	
\c@abspage	1015, 1754, 1761
\c@page	954, 1053, 1751, 2810, 2861
\c@zpage	1751, 1757
\c@zref@unique ..	915, 1746, 2916, 2918
\ch@pt@c	1909, 1915, 1921
\chapter	24, 30, 32, 61, 82
\ChapterPages	91, 112
\ChapterStart	78, 135, 150, 166
\ChapterStop	85, 148, 165, 184
\chardef	1159, 1174, 1183, 1187
\cleaders	2940
\cleardoublepage	79, 86
\clearpage	62
\closein	2136
\columnsep	1299
\columnwidth	1298
\comma@entry	343, 344, 346, 352,
	378, 379, 381, 387, 614, 616,
	620, 1616, 1617, 1618, 1624, 1627
\comma@parse	342, 377, 613, 1615

<code>\count@</code>	1333, 1344, 1345, 1347, 1382, 1383, 1392, 1394, 1395, 1508, 1531, 1533, 1534, 1539, 1540, 2062, 2067, 2078, 2079	<code>\endinput</code>	192, 264, 277, 966, 1004, 1025, 1047, 1088, 1128, 1219, 1237, 1411, 1456, 1583, 1694, 1788, 2032, 2435, 2447, 2455, 2468, 2539, 2544, 2895, 2950
<code>\csname</code>	252, 253, 290, 315, 316, 317, 326, 351, 352, 369, 370, 386, 387, 404, 405, 425, 427, 444, 462, 465, 478, 536, 538, 539, 544, 554, 559, 565, 578, 587, 604, 620, 651, 661, 730, 738, 744, 787, 788, 790, 810, 833, 834, 835, 896, 1323, 1484, 1645, 1653, 1705, 1734, 1736, 1739, 1741, 1753, 1759, 1764, 1765, 1767, 1769, 1770, 1777, 1830, 1833, 2071, 2078, 2090, 2114, 2218, 2220, 2235, 2236, 2253, 2260, 2263, 2277, 2278, 2298, 2395, 2418, 2420, 2590	<code>\escapechar</code>	329, 420, 460, 461, 467, 722, 1244
<code>\current@chapid</code>	80, 88	<code>\etex@unexpanded</code>	591, 819, 839, 2196
		<code>\evensidemargin</code>	1290
		<code>\externaldocument</code>	2222, 2264
D			
<code>\DeclareBoolOption</code>	2047, 2048, 2049, 2053	F	
<code>\DeclareOption</code>	194	<code>\fancyhead</code>	51, 54
<code>\default</code>	2367, 2407	<code>\fancyhf</code>	50, 53
<code>\define@key</code>	1829, 1832, 1835, 1838, 2050, 2900, 2903, 2906	<code>\fancypagestyle</code>	52
<code>\descriptionlabel</code>	1898	<code>\filename@area</code>	2197
<code>\detokenize</code>	1816	<code>\filename@parse</code>	2106
<code>\dftest</code>	167, 174, 175, 176, 177, 178, 179, 180, 181, 182	<code>\foo</code>	18, 29, 31, 33
<code>\dimen@</code>	2921, 2927, 2940	<code>\footskip</code>	1295
<code>\dimexpr</code>	153, 155, 1395, 2774, 2803, 2927	<code>\foremargin</code>	1302
<code>\directlua</code>	2449, 2473, 2481	<code>\frontmatter</code>	58, 103
<code>\do</code>	327, 720	G	
<code>\documentclass</code>	4, 39, 68, 272	<code>\g@addto@macro</code>	350, 368, 1739
<code>\dotfill</code>	169, 173	<code>\G@refundefinedtrue</code>	776
E		<code>\gdef</code>	412, 539, 544, 945, 1323, 1681, 1734, 1736
<code>\emph</code>	150	<code>\GetTitleStringDisableCommands</code>	1825
<code>\end</code>	34, 64, 130, 159, 183, 185	<code>\GetTitleStringExpand</code>	1811
<code>\endcsname</code>	252, 253, 290, 315, 316, 317, 326, 351, 352, 369, 370, 386, 387, 404, 405, 425, 427, 444, 463, 465, 478, 536, 538, 539, 544, 554, 559, 565, 578, 587, 604, 620, 651, 661, 736, 738, 744, 787, 788, 790, 810, 833, 834, 835, 883, 896, 1323, 1484, 1645, 1653, 1704, 1705, 1711, 1734, 1736, 1739, 1740, 1741, 1753, 1759, 1764, 1765, 1767, 1769, 1770, 1777, 1830, 1833, 2071, 2078, 2090, 2114, 2218, 2220, 2235, 2236, 2253, 2260, 2263, 2277, 2278, 2298, 2395, 2418, 2421, 2590	<code>\GetTitleStringNonExpand</code>	1813
		<code>\GetTitleStringResult</code>	1816
		H	
		<code>\hb@xt@</code>	2940
		<code>\headheight</code>	1293
		<code>\headmargin</code>	1305
		<code>\headsep</code>	1294
		<code>\hfill</code>	2928, 2940
		<code>\hoffset</code>	1286
		<code>\hss</code>	2940
I			
<code>\if@filesw</code>	710, 1051, 1679, 2483, 2489, 2496	<code>\if@safe@actives</code>	889
<code>\ifcase</code>	115, 1195, 1616, 2687, 2700, 2764, 2793	<code>\ifcsname</code>	883, 1704, 1711, 1740
<code>\ifcsw</code>	883, 1704, 1711, 1740	<code>\ifeof</code>	2121, 2175
<code>\ifetex@unexpanded</code>	267	<code>\ifheadnameref</code>	1917, 1930
<code>\ifheadnameref</code>	1917, 1930	<code>\ifin@</code>	317
<code>\ifluatex</code>	1263, 1415, 1493, 1506, 1549, 1565, 2672, 2698, 2753, 2783	<code>\ifnum</code>	476, 1065, 1170, 1180, 1186, 1382, 1539, 1588, 1641, 2067, 2144, 2148, 2153, 2462, 2501, 2564, 2576, 2619, 2632, 2650, 2760, 2789, 2925

<code>\ifodd</code>	124, 2810, 2861	<code>\ltx@firstofone</code>	254, 867, 878, 884, 1465, 1466
<code>\ifpdf</code> ...	2458, 2749, 2754, 2779, 2784	<code>\ltx@firstoftwo</code>	799, 826, 827, 892, 1067, 2652, 2654	
<code>\ifx</code> 437, 441, 474, 507, 565, 673, 676,	690, 729, 795, 964, 969, 976,	<code>\ltx@firsttwo</code>	2643	
	1002, 1023, 1045, 1086, 1126,	<code>\ltx@gobble</code>	250, 355, 390, 623, 663, 969,
	1217, 1235, 1322, 1409, 1455,		970, 976, 1385, 1542, 1629, 1850	
	1464, 1483, 1494, 1550, 1581,	<code>\ltx@gobblethree</code>	977	
	1598, 1607, 1611, 1616, 1617,	<code>\ltx@gobbletwo</code>	694, 911, 1008, 1631, 1716
	1618, 1692, 1764, 1786, 1909,	<code>\ltx@ifpackageloaded</code>	2315	
	1913, 1970, 1990, 2014, 2030,	<code>\ltx@ifUndefined</code> ..	229, 249, 257,	410, 877, 919, 1103, 1430, 1442,
	2078, 2161, 2184, 2189, 2194,		1462, 1463, 1492, 1548, 1904,	1906, 2450, 2460, 2499, 2748, 2778
	2208, 2255, 2346, 2349, 2374,	<code>\ltx@ifundefined</code>	300, 485, 583, 760, 805,
	2377, 2384, 2388, 2404, 2433,		940, 1245, 1317, 1478, 2040,	2300, 2301, 2587, 2658, 2659,
	2445, 2449, 2473, 2481, 2537,		2660, 2676, 2677, 2678, 2688, 2701	
	2542, 2759, 2788, 2893, 2948, 2955	<code>\ltx@LocalAppendToMacro</code>	385, 403, 649, 659, 1562,
<code>\ifZREF@found</code>	247, 2365, 2372		2219, 2259, 2262, 2367, 2370,	2379, 2386, 2392, 2399, 2406, 2418
<code>\ifZREF@immediate</code>	<code>\ltx@newif</code>	1310, 1471	
	634, 700, 712, 716, 731	<code>\ltx@one</code>	1383, 1540,	1746, 2062, 2068, 2761, 2790, 2916
<code>\ifZREF@pa@list</code>	1471, 1476	<code>\ltx@onelevel@sanitize</code>	557, 562	
<code>\ifZREF@pl@list</code>	1310, 1315	<code>\ltx@ReturnAfterFi</code>	2351	
<code>\ifzref@titleref@expand</code> ..	1794, 1810	<code>\ltx@ssecondoftwo</code>	311,	784, 797, 827, 886, 890, 1069, 2652
<code>\ifzref@titleref@stripperperiod</code>	<code>\ltx@space</code>	584, 586, 806, 815, 829,	832, 1180, 1186, 1346, 1393,
	1807, 1818		1590, 1618, 1661, 2553, 2560, 2623	
<code>\ifZREF@xr@toltxlabel</code> ...	2241, 2283	<code>\ltx@two</code>	2747, 2759, 2777, 2788	
<code>\ifZREF@xr@tozreflabel</code> ..	2227, 2269	<code>\ltx@zero</code> ..	476, 1588, 1641, 1722,	2058, 2501, 2564, 2576, 2619,
<code>\ifZREF@xr@curluse</code> ..	2111, 2390, 2420		2632, 2650, 2750, 2755, 2780, 2785	
<code>\ifZREF@xr@verbose</code> ..	2229, 2271, 2292			
<code>\ifZREF@xr@zreflabel</code>			
	2042, 2128, 2142, 2183			
<code>\immediate</code>	705, 1680			
<code>\in@</code>	314			
<code>\item</code>	107, 111, 113, 121, 125, 127			
K				
<code>\kern</code>	2936			
<code>\kv@define@key</code>	644, 655, 666, 671, 688			
<code>\kv@key</code> ...	692, 1610, 1612, 1613, 1627			
<code>\kv@parse</code>	689, 1606			
<code>\kv@value</code>	690, 1607, 1608, 1615			
<code>\kvsetkeys</code>	633, 1842, 2055, 2912			
L				
<code>\label</code>	969, 976, 1850	<code>\m@ne</code>	1053	
<code>\lastxpos</code>	2477	<code>\M@sect</code>	1929	
<code>\lastypos</code>	2478	<code>\M@TitleReference</code>	2309	
<code>\leavevmode</code>	2915	<code>\mag</code>	1256, 2760, 2774, 2789, 2803	
<code>\lst@caption</code>	1993	<code>\mainmatter</code>	60, 134	
<code>\lst@label</code>	1990	<code>\makeatletter</code>	11, 74, 101, 2091	
<code>\lst@MakeCaption</code>	1989	<code>\makeatother</code>	16, 99	
<code>\LT@c@ption</code>	1981	<code>\makebox</code>	169, 170	
<code>\ltx@backslashchar</code>	<code>\MakeRobustcommand</code>	232	
	743, 1505, 1564, 2073	<code>\marginparsep</code>	1297	
<code>\ltx@car</code>	1610, 2164	<code>\marginparwidth</code>	1296	
<code>\ltx@cdr</code>	1612, 2167	<code>\meaning</code>	2011	
<code>\ltx@empty</code>	290, 499,	<code>\mediaheight</code>	1262	
	565, 632, 719, 796, 1095, 1598,	<code>\mediawidth</code>	1261	
	1795, 1821, 1828, 1909, 1913,	<code>\MessageBreak</code> ..	270, 513, 564, 570,	680, 1332, 1344, 1348, 1396,
	1970, 1990, 2103, 2138, 2161,		1507, 1532, 1535, 1566, 1568,	
	2167, 2208, 2362, 2955, 2958, 2961			

<code>\TeXeTstate</code>	2501		Y	
<code>\the</code>	13, 153, 155, 429, 444, 460, 556, 561, 620, 626, 742, 756, 922, 1015, 1059, 1100, 1106, 1167, 1316, 1336, 1343, 1344, 1345, 1347, 1392, 1394, 1395, 1423, 1425, 1438, 1450, 1477, 1512, 1530, 1531, 1533, 1804, 2197, 2203, 2214, 2248, 2333, 2342, 2474, 2475, 2477, 2478		<code>\y</code>	2010, 2011
<code>\thechapter</code>	14		Z	
<code>\thefoo</code>	7, 12, 20		<code>\z</code>	2011, 2012, 2014
<code>\theotype</code>	2260		<code>\z@</code>	2936
<code>\thepage</code>	43, 44, 45, 713, 717, 778, 936, 1752		<code>\Z@D@page</code>	1154
<code>\thezpage</code>	18, 1752, 1758		<code>\Z@L@LastPage</code>	1056
<code>\thezref@unique</code>	11, 914, 1749, 1750, 1757, 1758, 1760, 2920, 2922, 2925, 2926, 2934		<code>\Z@L@main</code>	1055
<code>\title</code>	2379, 2408		<code>\Z@L@ZREF@temp</code>	632, 636, 639, 650, 660
<code>\toks@</code>	423, 429, 443, 444, 553, 556, 558, 561, 612, 619, 620, 626, 740, 742, 755, 756, 1054, 1059, 1332, 1336, 1342, 1343, 1505, 1512, 1529, 1530, 1798, 1804, 2182, 2197		<code>\zdotfill</code>	20, 170, 173, 2913
<code>\topmargin</code>	1288		<code>\zdotfillsetup</code>	20, 2912
<code>\TR@TitleReference</code> ..	2304, 2364, 2403		<code>\zexternaldocument</code>	21, 2085
<code>\trimedge</code>	1300		<code>\ziflastpage</code>	13, 1072
<code>\trimtop</code>	1303		<code>\zifrefundefined</code>	8, 762
<code>\ttl@sect@i</code>	1962		<code>\zkvlabel</code>	975
<code>\ttl@straight@i</code>	1968		<code>\zlabel</code>	12, 83, 104, 138, 146, 968
			<code>\zlistpageattr</code>	1472
U			<code>\zlistpagelayout</code>	17, 1311
<code>\unexpanded</code>	270, 275		<code>\zmakeperpage</code>	18, 1720
<code>\UniqueCounterCall</code>	1151		<code>\znexpage</code>	15, 51, 54, 1150
<code>\UniqueCounterNew</code>	1134		<code>\znexpagesetup</code>	15, 42, 1136
<code>\uppermargin</code>	1304		<code>\znonextpagename</code>	46, 1153, 1201
<code>\url</code>	2399		<code>\zpageref</code>	12, 126, 990
<code>\urluse</code>	2394		<code>\zposx</code>	19, 153, 2524, 2925, 2926
<code>\usepackage</code>	9, 41, 48, 70, 72		<code>\zposy</code>	19, 155, 2524
			<code>\zref</code>	12, 25, 26, 27, 28, 112, 114, 123, 128, 129, 139, 981, 991
V			<code>\ZREF@@@delprop</code> ...	434, 436, 471, 473
<code>\value</code>	13, 1100, 1316, 1477		<code>\ZREF@@@newprop</code>	538, 542
<code>\verb</code>	173		<code>\ZREF@@delprop</code>	424, 433, 447, 464, 470, 480
<code>\voffset</code>	1287		<code>\ZREF@@extract</code>	808, 814
			<code>\ZREF@@makeperpage</code> ..	1722, 1728, 1732
W			<code>\ZREF@@newprop</code>	516, 527, 530, 534
<code>\write</code>	704, 705, 1680		<code>\ZREF@@perpage@step</code>	1737, 1745
			<code>\ZREF@abspos@body@x@center</code>	2840
X			<code>\ZREF@abspos@body@x@left</code> .	2838, 2865
<code>\x</code>	330, 335, 723, 728, 897, 899, 1193, 1209, 1246, 1253, 1335, 1338, 1341, 1381, 1510, 1515, 1528, 1538, 1643, 1648, 1651, 1657, 1727, 1730, 2008, 2009, 2014, 2181, 2184, 2189, 2194, 2204, 2207, 2217, 2218, 2249, 2254		<code>\ZREF@abspos@body@x@right</code>	2839, 2862
<code>\XR@ext</code>	2040		<code>\ZREF@abspos@body@y@bottom</code>	2845, 2857, 2879
			<code>\ZREF@abspos@body@y@center</code>	2849, 2880
			<code>\ZREF@abspos@body@y@top</code>	2841, 2846, 2850, 2878
			<code>\ZREF@abspos@foot@x@center</code>	2855
			<code>\ZREF@abspos@foot@x@left</code>	2853
			<code>\ZREF@abspos@foot@x@right</code>	2854
			<code>\ZREF@abspos@foot@y@bottom</code>	2856
			<code>\ZREF@abspos@head@x@center</code>	2820, 2840, 2855
			<code>\ZREF@abspos@head@x@left</code>	2806, 2817, 2821, 2838, 2853
			<code>\ZREF@abspos@head@x@right</code>	2816, 2839, 2854
			<code>\ZREF@abspos@head@y@bottom</code>	2830, 2842
			<code>\ZREF@abspos@head@y@center</code>	2834
			<code>\ZREF@abspos@head@y@top</code>	2824, 2831, 2835

<code>\ZREF@abspos@marginpar@x@center</code>	2874	<code>\zref@def@absposnumx</code>	2599
<code>\ZREF@abspos@marginpar@x@left</code>	2860 , 2871 , 2875	<code>\ZREF@def@absposnumy</code>	2605
<code>\ZREF@abspos@marginpar@x@right</code>	2870	<code>\zref@def@absposnumy</code>	2602
<code>\ZREF@abspos@marginpar@y@bottom</code>	2879	<code>\zref@def@absposx</code>	2593
<code>\ZREF@abspos@marginpar@y@center</code>	2880	<code>\zref@def@absposy</code>	2596
<code>\ZREF@abspos@marginpar@y@top</code>	2878	<code>\ZREF@def@extract</code>	845 , 847
<code>\ZREF@abspos@media@height</code>	2675 , 2721 , 2727	<code>\zref@def@extract</code>	8 , 844
<code>\ZREF@abspos@media@width</code>	2657 , 2714 , 2718	<code>\ZREF@def@extractdefault</code>	856 , 858
<code>\ZREF@abspos@media@x@center</code>	2716	<code>\zref@def@extractdefault</code>	855
<code>\ZREF@abspos@media@x@left</code>	2710 , 2717	<code>\ZREF@default</code>	561 , 562 , 571
<code>\ZREF@abspos@media@x@right</code>	2713	<code>\zref@default</code>	9 , 530 , 806 , 925 , 927
<code>\ZREF@abspos@media@y@bottom</code>	2723	<code>\ZREF@delprop</code>	412 , 415 , 417 , 452 , 455 , 457
<code>\ZREF@abspos@media@y@center</code>	2726	<code>\zref@delprop</code>	411 , 451
<code>\ZREF@abspos@media@y@top</code>	2720 , 2738	<code>\ZREF@df@dot</code>	2907 , 2911 , 2940
<code>\ZREF@abspos@origin@x</code>	2747 , 2808	<code>\ZREF@df@min</code>	2904 , 2909 , 2927
<code>\ZREF@abspos@origin@y</code>	2777 , 2826	<code>\ZREF@df@unit</code>	2901 , 2910 , 2921
<code>\ZREF@abspos@paper@x@center</code>	2735 , 2883	<code>\ZREF@dotfill</code>	2923 , 2930 , 2939
<code>\ZREF@abspos@paper@x@left</code>	2729 , 2807 , 2881	<code>\ZREF@ENV@line</code>	2958 , 2961
<code>\ZREF@abspos@paper@x@right</code>	2732 , 2882	<code>\zref@env@line</code>	2953 , 2954
<code>\ZREF@abspos@paper@y@bottom</code>	2739 , 2885	<code>\ZREF@extract</code>	803 , 820 , 823 , 875
<code>\ZREF@abspos@paper@y@center</code>	2743 , 2886	<code>\zref@extract</code>	8 , 95 , 96 , 109 , 140 , 803 , 823 , 852 , 870 , 875 , 988 , 1109 , 1205 , 1347 , 1394 , 1395 , 1519 , 1556 , 1757 , 1758 , 1851 , 2525 , 2528 , 2714 , 2718 , 2721 , 2727 , 2733 , 2736 , 2741 , 2745 , 2766 , 2795 , 2809 , 2811 , 2813 , 2818 , 2822 , 2827 , 2828 , 2832 , 2836 , 2843 , 2847 , 2851 , 2858 , 2863 , 2866 , 2867 , 2872 , 2876
<code>\ZREF@abspos@paper@y@top</code>	2738 , 2740 , 2744 , 2825 , 2884	<code>\ZREF@extractdefault</code>	824 , 840 , 843 , 874
<code>\ZREF@abspos@stock@x@center</code>	2883	<code>\zref@extractdefault</code>	8 , 116 , 117 , 816 , 843 , 863 , 869 , 874 , 1065 , 1066 , 1163 , 1178 , 1224 , 1760 , 2302 , 2305 , 2306 , 2310 , 2311 , 2314 , 2316 , 2317 , 2319 , 2321 , 2551 , 2558 , 2617 , 2645 , 2810 , 2861
<code>\ZREF@abspos@stock@x@left</code>	2881	<code>\ZREF@false</code>	676 , 686
<code>\ZREF@abspos@stock@x@right</code>	2882	<code>\ZREF@foundfalse</code>	2363
<code>\ZREF@abspos@stock@y@bottom</code>	2885	<code>\ZREF@foundtrue</code>	2410
<code>\ZREF@abspos@stock@y@center</code>	2886	<code>\ZREF@getcurrent</code>	581 , 592 , 595 , 873
<code>\ZREF@abspos@stock@y@top</code>	2884	<code>\zref@getcurrent</code>	7 , 595 , 868 , 873
<code>\ZREF@abspos@used</code>	2609 , 2611	<code>\zref@hex</code>	1423 , 1425 , 1438 , 1450 , 1465 , 1468
<code>\ZREF@absposnum</code>	2568 , 2580 , 2586	<code>\zref@ifabsposnumundefined</code>	2644 , 2649
<code>\zref@absposnumused</code>	2631	<code>\zref@ifabsposundefined</code>	2642
<code>\zref@absposnumx</code>	2550 , 2562 , 2600	<code>\ZREF@ifDefinable</code>	241 , 762 , 990 , 993 , 1072 , 1114 , 1150 , 1311 , 1472 , 1720 , 1774 , 1841 , 1844 , 2085 , 2509 , 2514 , 2519 , 2913
<code>\zref@absposnumy</code>	2557 , 2574 , 2603	<code>\ZREF@iflastpage</code>	1073 , 1075 , 1075
<code>\zref@absposused</code>	2608	<code>\zref@iflastpage</code>	13 , 1064 , 1078
<code>\zref@absposx</code>	2548 , 2594	<code>\zref@iflistcontainsprop</code>	6 , 309 , 344 , 362 , 379 , 397 , 647 , 657
<code>\zref@absposy</code>	2555 , 2597	<code>\zref@iflistundefined</code>	6 , 288 , 299 , 303 , 310
<code>\zref@addprop</code>	6 , 76 , 359 , 1016 , 1028 , 1093 , 1096 , 1251 , 1267 , 1271 , 1275 , 1279 , 1424 , 1426 , 1439 , 1451 , 1627 , 1793 , 2438 , 2547		
<code>\zref@addprops</code>	6 , 15 , 340 , 937 , 1309 , 1699 , 2480		
<code>\ZREF@addtoks</code>	754		
<code>\ZREF@base@ok</code>	957 , 964 , 1002 , 1023 , 1045 , 1086 , 1126 , 1217 , 1235 , 1409 , 1581 , 1692 , 1786 , 2030 , 2433 , 2445 , 2537 , 2893 , 2948		
<code>\ZREF@call</code>	1159 , 1174 , 1183 , 1187 , 1195		
<code>\ZREF@def@abspos</code>	2594 , 2597 , 2600 , 2603 , 2605		

<code>\zref@ifpropundefined</code>	7, 484 , 488 , 518 , 547 , 614 , 827 , 1390 , 1640 , 2357	<code>\ZREF@nextpage</code>	1151 , 1155
<code>\ZREF@ifrefcontainsprop</code>	... 786 , 794	<code>\ZREF@nil</code>	544 , 796 , 835 , 2174 , 2180 , 2186 , 2191 , 2201 , 2217 , 2246 , 2254 , 2345 , 2352 , 2361 , 2364 , 2403
<code>\zref@ifrefcontainsprop</code>	... 9 , 782 , 1392 , 2318 , 2414 , 2415	<code>\ZREF@NOVALUE</code>	802
<code>\ZREF@ifrefundefined</code>	... 764 , 767 , 1160 , 1171 , 1181	<code>\ZREF@novalue</code>	795 , 796 , 802
<code>\zref@ifrefundefined</code>	... 8 , 759 , 769 , 775 , 783 , 826 , 1172 , 1345 , 1533 , 1750 , 2565 , 2577 , 2613 , 2643 , 2651 , 2922	<code>\ZREF@np@call@next</code>	1145 , 1149 , 1204
<code>\ZREF@immediatetrue</code>	677	<code>\ZREF@np@call@nonext</code>	1142 , 1148 , 1200
<code>\ZREF@label</code>	602 , 626 , 636 , 639 , 709 , 1059	<code>\ZREF@np@call@unknown</code>	1138 , 1147 , 1196
<code>\zref@label</code>	... 7 , 596 , 972	<code>\ZREF@np@setup@i</code>	1137 , 1140
<code>\zref@labelbykv</code>	... 629 , 979	<code>\ZREF@np@setup@ii</code>	1141 , 1144
<code>\zref@labelbylist</code>	... 7 , 597 , 599 , 1100 , 1749 , 2511	<code>\ZREF@number</code>	919 , 1519 , 1525 , 1587 , 2650
<code>\zref@labelbyprops</code>	... 7 , 88 , 609 , 1158 , 2516 , 2521 , 2920 , 2934	<code>\ZREF@org@begintheorem</code>	2018
<code>\zref@listexists</code>	... 6 , 302 , 321 , 341 , 360 , 376 , 395 , 418 , 458 , 601	<code>\ZREF@org@caption</code>	1858
<code>\zref@listforloop</code>	... 320 , 656	<code>\ZREF@org@chapter</code>	1870 , 1926
<code>\zref@listpageattr</code>	... 1472	<code>\ZREF@org@opargbegintheorem</code>	2003
<code>\zref@listpagelayout</code>	... 1311	<code>\ZREF@org@part</code>	1864
<code>\zref@localaddprop</code>	... 394	<code>\ZREF@org@schapter</code>	1888
<code>\zref@localaddprops</code>	... 375	<code>\ZREF@org@ssect</code>	1876
<code>\zref@localdelprop</code>	... 414 , 454 , 668	<code>\ZREF@org@spart</code>	1882
<code>\ZREF@mainlist</code>	... 597 , 931 , 934 , 937 , 1016 , 1028 , 1793 , 2438	<code>\ZREF@org@sssect</code>	1894
<code>\ZREF@makeperpage@opt</code>	... 1722 , 1725	<code>\ZREF@org@stpelt</code>	1709 , 1714 , 1718
<code>\ZREF@MARKS@DefineProp</code>	... 1603 , 1604 , 1605 , 1639	<code>\ZREF@org@beamer@section</code>	1944
<code>\zref@marks@register</code>	... 1585 , 1590 , 1622 , 1661	<code>\ZREF@org@beamer@subsection</code>	1950
<code>\ZREF@name</code>	... 228 , 258 , 269 , 291 , 294 , 304 , 345 , 363 , 380 , 398 , 489 , 508 , 520 , 524 , 548 , 563 , 615 , 679 , 691 , 1589 , 2451 , 2463	<code>\ZREF@org@beamer@subsubsection</code>	1956
<code>\ZREF@NAME@bot</code>	... 1618 , 1638	<code>\ZREF@org@descriptionlabel</code>	1900
<code>\ZREF@NAME@first</code>	... 1617 , 1637	<code>\ZREF@org@lst@MakeCaption</code>	1996
<code>\ZREF@NAME@top</code>	... 1616 , 1636	<code>\ZREF@org@LT@cOption</code>	1982
<code>\zref@newlabel</code>	... 8 , 281 , 284 , 749 , 2201 , 2287	<code>\ZREF@org@M@ssect</code>	1935
<code>\zref@newlist</code>	... 6 , 287 , 934 , 1049 , 1092 , 1613 , 1698 , 2472	<code>\ZREF@org@refstepcounter</code>	1033
<code>\ZREF@newprop</code>	... 497 , 500 , 503	<code>\ZREF@org@stepcounter</code>	1702 , 1707 , 1713
<code>\zref@newprop</code>	7 , 12 , 13 , 14 , 75 , 494 , 935 , 936 , 939 , 946 , 950 , 954 , 1015 , 1027 , 1248 , 1264 , 1268 , 1272 , 1276 , 1306 , 1307 , 1308 , 1423 , 1425 , 1438 , 1450 , 1644 , 1652 , 2036 , 2037 , 2038 , 2358 , 2474 , 2475 , 2477 , 2478 , 2952 , 2953	<code>\ZREF@org@testdef</code>	... 1318 , 1320 , 1479 , 1481
<code>\ZREF@NewPropAnchor</code>	... 938 , 2087 , 2437	<code>\ZREF@org@thepage</code>	713 , 717
<code>\ZREF@NewPropPageValue</code>	... 953 , 1094 , 1697	<code>\ZREF@org@ttl@ssect@i</code>	1964
<code>\ZREF@NewPropTheotype</code>	... 949 , 2258	<code>\ZREF@org@ttl@straight@i</code>	1975
<code>\ZREF@NewPropTitle</code>	... 944 , 1792 , 2088	<code>\ZREF@org@write</code>	704 , 705
		<code>\ZREF@P</code>	504 , 505 , 507 , 509 , 518 , 521 , 525 , 535 , 536 , 538 , 539 , 540 , 544 , 720 , 724 , 725 , 734 , 738 , 743 , 744
		<code>\ZREF@pa@AfterLastShipout</code>	1475 , 1573
		<code>\ZREF@pa@AtVeryEnd</code>	1488 , 1491 , 1562
		<code>\ZREF@pa@ListPage</code>	1509 , 1527
		<code>\ZREF@pa@listtrue</code>	1473
		<code>\ZREF@page@max</code>	1316 , 1382 , 1477 , 1539
		<code>\zref@pageattr</code>	... 1517
		<code>\zref@pageattr@used</code>	... 1524
		<code>\ZREF@pagenum@last</code>	1177 , 1180
		<code>\ZREF@pagenum@this</code>	... 1162 , 1167 , 1170 , 1180 , 1186
		<code>\ZREF@par</code>	507 , 532
		<code>\ZREF@param</code>	... 421 , 422 , 441 , 459 , 476 , 645 , 646 , 647 , 651 , 672 , 673 , 676 , 681
		<code>\ZREF@patch</code>	248 , 1030 , 1855 , 1861 , 1867 , 1873 , 1879 , 1885 , 1891 ,

1897, 1928, 1941, 1947, 1953,
 1961, 1967, 1980, 1988, 2000, 2015
 \zref@pdfpageattr 1434, 1517, 1523, 1534
 \zref@pdfpageattr@used 1435
 \zref@pdfpagesattr .. 1446, 1554, 1567
 \zref@pdfpagesattr@used . 1447, 1559
 \ZREF@pl@AfterLastShipout 1314, 1401
 \ZREF@pl@AtVeryEnd 1327, 1330
 \ZREF@pl@ListEntry
 . 1349, 1350, 1351, 1352, 1353,
 1354, 1355, 1356, 1357, 1358,
 1359, 1360, 1361, 1362, 1363,
 1364, 1365, 1366, 1367, 1368,
 1369, 1370, 1371, 1372, 1373,
 1374, 1375, 1376, 1377, 1378, 1389
 \ZREF@pl@ListPage 1334, 1340
 \ZREF@pl@Listtrue 1312
 \zref@pos@label@used 2623
 \zref@pos@num@used 2637
 \zref@prop 323, 331, 332, 336, 657, 661
 \zref@propexists 7, 343, 361,
 378, 396, 487, 577, 646, 667, 982
 \ZREF@refname@next
 1165, 1172, 1181, 1205
 \ZREF@refname@this
 1157, 1158, 1160, 1163
 \ZREF@RefPrefix . 283, 285, 1322, 1483
 \ZREF@refused 772, 774
 \zref@refused
 ... 8, 768, 771, 848, 859, 987,
 994, 1076, 1077, 1112, 1227,
 1525, 1560, 1849, 2612, 2620, 2633
 \zref@require@unique
 11, 908, 1701, 2898
 \ZREF@Robust 231,
 237, 243, 284, 287, 302, 309,
 340, 359, 375, 394, 411, 414,
 451, 454, 487, 494, 546, 576,
 596, 599, 609, 629, 701, 771,
 844, 855, 866, 882, 908, 924,
 930, 1111, 1524, 1559, 1796,
 1809, 2593, 2596, 2599, 2602, 2608
 \ZREF@SavedEscapechar 460, 467
 \zref@savepos ... 20, 2481, 2497, 2503
 \ZREF@savepos@ok 2530, 2542
 \zref@setcurrent
 7, 81, 540, 576, 696, 1032
 \zref@setdefault 9, 924, 927
 \zref@setmainlist 9, 930
 \zref@showprop 546
 \ZREF@STAR 1611, 1635
 \ZREF@stripperperiod 1820, 1828
 \ZREF@temp 193, 200, 201, 202,
 203, 204, 205, 206, 207, 208,
 209, 210, 211, 212, 213, 214,
 215, 216, 232, 233, 439, 440,
 441, 719, 740, 741, 749, 1242,
 1256, 1257, 1258, 1259, 1260,
 1261, 1262, 1281, 1282, 1283,
 1284, 1286, 1287, 1288, 1289,
 1290, 1291, 1292, 1293, 1294,
 1295, 1296, 1297, 1298, 1299,
 1300, 1301, 1302, 1303, 1304,
 1305, 1321, 1322, 1414, 1427,
 1440, 1452, 1455, 1461, 1462,
 1463, 1464, 1482, 1483, 1491,
 1492, 1493, 1494, 1547, 1548,
 1549, 1550, 1610, 1611, 1969,
 1970, 2061, 2071, 2074, 2078,
 2616, 2619, 2620, 2747, 2750,
 2755, 2759, 2761, 2764, 2777,
 2780, 2785, 2788, 2790, 2793, 2918
 \ZREF@TempName .. 1586, 1598, 1599,
 1601, 1627, 1640, 1644, 1652, 1663
 \ZREF@TempNum
 1587, 1588, 1592, 1599, 1641, 1654
 \zref@thepage 14, 1108, 1117
 \zref@thepage@atbegshi@hook
 1095, 1099
 \zref@thepage@name
 14, 1103, 1109, 1112, 1166
 \zref@thepage@refused ... 1111, 1116
 \ZREF@titleref 1845, 1847
 \zref@titleref@cleanup .. 1796, 1836
 \zref@titleref@current
 944, 1815, 1819, 1820, 1839
 \ZREF@titleref@hook
 1795, 1799, 1803, 1826
 \zref@titleref@setcurrent
 . 1809, 1857, 1863, 1869, 1875,
 1881, 1887, 1893, 1899, 1907,
 1910, 1914, 1918, 1920, 1931,
 1933, 1943, 1949, 1955, 1963,
 1971, 1973, 1983, 1992, 2002, 2017
 \zref@titleref@stripperperiodtrue 1808
 \ZREF@true 673, 687
 \ZREF@u@getcurrent 590
 \zref@unhex 1466, 1469, 1518, 1555
 \ZREF@UpdatePdfTeX ... 246, 2454, 2467
 \ZREF@value 556, 557, 570
 \ZREF@wrapper@babel 899, 905
 \zref@wrapper@babel
 11, 140, 764, 772, 845,
 856, 882, 972, 979, 983, 1073,
 1845, 2594, 2597, 2600, 2603, 2609
 \zref@wrapper@immediate
 11, 87, 635, 701, 1058
 \ZREF@wrapper@unexpanded ... 866, 880
 \zref@wrapper@unexpanded
 11, 867, 872, 877, 2297
 \ZREF@wu@extract 818, 870
 \ZREF@wu@extractdefault ... 838, 869
 \ZREF@wu@getcurrent 590, 868
 \ZREF@X 496, 499, 536
 \zref@xr@ 2051

<code>\ZREF@xr@@AddUrl</code>	2063, 2066	<code>\ZREF@xr@relax</code>	2290, 2377
<code>\ZREF@xr@@input</code>	2194, 2289	<code>\ZREF@xr@scanparams</code>	2252, 2361
<code>\ZREF@xr@AddURL</code>	2059, 2112, 2391	<code>\ZREF@xr@scantitleref</code> ...	2364, 2403
<code>\ZREF@xr@checkfile</code> ..	2116, 2119, 2169	<code>\ZREF@xr@temp</code>	2376, 2377
<code>\ZREF@xr@checkkey</code>	2347, 2356	<code>\ZREF@xr@tempname</code> 2205, 2206, 2226,	
<code>\ZREF@xr@checklist</code>	2217, 2345	2231, 2242, 2250, 2251, 2268, 2284	
<code>\zref@xr@ext</code>	22, 2039, 2107	<code>\ZREF@xr@temprefname</code>	
<code>\ZREF@xr@externaldocument</code>		2206, 2218, 2220,	
2094, 2097, 2101		2236, 2251, 2253, 2260, 2263, 2278	
<code>\ZREF@xr@externalfile</code>		<code>\ZREF@xr@theURL</code>	
2104, 2105, 2223, 2265		2069, 2071, 2073, 2079, 2114, 2395	
<code>\ZREF@xr@file</code> ...	2105, 2120, 2123,	<code>\ZREF@xr@tolabel</code> ...	2242, 2284, 2291
2129, 2140, 2163, 2211, 2330, 2339		<code>\ZREF@xr@URL</code> ...	2057, 2067, 2068, 2069
<code>\ZREF@xr@filelist</code>	2103,	<code>\ZREF@xr@url</code> ...	2110, 2112, 2113, 2421
2161, 2164, 2166, 2167, 2195, 2196		<code>\ZREF@xr@urlcheck</code> ..	2226, 2268, 2413
<code>\ZREF@xr@found</code> .	2131, 2141, 2203, 2248	<code>\ZREF@xr@zref@ignorewarning</code>	
<code>\ZREF@xr@graburl</code>	2107, 2109	2238, 2280, 2327	
<code>\ZREF@xr@ignored@empty</code>		<code>\ZREF@xr@zref@newlabel</code> ..	2184, 2287
2132, 2144, 2146, 2213, 2214		<code>\ZREF@xr@zreflabelfalse</code>	2093
<code>\ZREF@xr@ignored@ltx</code>		<code>\ZREF@xr@zreflabeltrue</code>	2096
2134, 2153, 2155, 2341, 2342		<code>\ZREF@zref</code>	983, 986
<code>\ZREF@xr@ignored@zref</code>		<code>\ZREF@zsavepos</code> .	2494, 2511, 2516, 2521
2133, 2148, 2150, 2332, 2333		<code>\zrefused</code> 12, 92, 93, 161, 162, 163, 993	
<code>\ZREF@xr@line</code> ..	2173, 2174, 2186, 2191	<code>\zruns</code>	17, 1673
<code>\ZREF@xr@list</code>	2207, 2208	<code>\zsavepos</code>	19, 157, 158, 2509
<code>\ZREF@xr@ltx@ignorewarning</code>	2336	<code>\zsaveposx</code>	19, 2514
<code>\ZREF@xr@newlabel</code>	2189, 2288	<code>\zsaveposy</code>	2519
<code>\ZREF@xr@prefix</code>	2102, 2202,	<code>\zthepage</code>	14, 1114
2238, 2242, 2247, 2273, 2280, 2284		<code>\ztitleref</code>	18, 1844
<code>\ZREF@xr@process@label</code> ..	2191, 2246	<code>\ztitlerefsetup</code>	19, 1829
<code>\ZREF@xr@process@zreflabel</code> 2186, 2201		<code>\ztotpages</code>	16, 124, 1223
<code>\ZREF@xr@processfile</code>	2119, 2172	<code>\zunknownnextpagename</code> .	15, 1154, 1197
<code>\ZREF@xr@processline</code>	2174, 2180	<code>\zunmakeperpage</code>	18, 1774
<code>\ZREF@xr@refname</code>		<code>\zxrsetup</code>	21, 2054
2202, 2228, 2235, 2247, 2270, 2277			