

The zref package

Heiko Oberdiek*

<heiko.oberdiek at gmail.com>

2016/05/16 v2.25

Abstract

Package `zref` tries to get rid of the restriction in L^AT_EX'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 L ^A T _E X behaviour	4
1.2	Basic idea	5
1.3	Interfaces	5
2	Interface for programmers	5
2.1	Entities	6
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

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

3.9	Module <code>runs</code>	17
3.10	Module <code>perpage</code>	17
3.11	Module <code>counter</code>	18
3.12	Module <code>titleref</code>	18
3.13	Module <code>savepos</code>	19
3.14	Module <code>dotfill</code>	20
3.15	Module <code>env</code>	20
3.16	Module <code>xr</code>	20
4	ToDo	21
5	Example	21
6	Implementation	24
6.1	Package <code>zref</code>	24
6.1.1	Identification	24
6.1.2	Load basic module	24
6.1.3	Process options	24
6.2	Module <code>base</code>	25
6.2.1	Prefixes	25
6.2.2	Identification	25
6.2.3	Utilities	25
6.2.4	Check for ε -TeX	26
6.2.5	Auxiliary file stuff	27
6.2.6	Property lists	27
6.2.7	Properties	31
6.2.8	Reference generation	34
6.2.9	Reference querying and extracting	37
6.2.10	Compatibility with <code>babel</code>	41
6.2.11	Unique counter support	41
6.2.12	Utilities	41
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>	44
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>	48
6.10	Module <code>pagelayout</code>	49
6.10.1	Support for LuaTeX	49
6.10.2	Define layout properties	50
6.11	Module <code>pageattr</code>	53
6.12	Module <code>marks</code>	56
6.13	Module <code>runs</code>	57
6.14	Module <code>perpage</code>	58
6.15	Module <code>titleref</code>	60
6.15.1	Implementation	60
6.15.2	User interface	62
6.15.3	Patches for section and caption commands	62
6.15.4	Environment <code>description</code>	63
6.15.5	Class <code>memoir</code>	63
6.15.6	Class <code>beamer</code>	64
6.15.7	Package <code>titlesec</code>	64

6.15.8	Package <code>longtable</code>	65
6.15.9	Package <code>listings</code>	65
6.15.10	Theorems	65
6.16	Module <code>xr</code>	66
6.17	Module <code>hyperref</code>	74
6.18	Module <code>savepos</code>	74
6.18.1	Identification	74
6.18.2	Availability	74
6.18.3	Setup	75
6.18.4	User macros	75
6.19	Module <code>abspos</code>	76
6.19.1	Identification	76
6.19.2	Media	79
6.19.3	Paper	80
6.19.4	Origin	81
6.19.5	Header	81
6.19.6	Body	82
6.19.7	Footer	83
6.19.8	Marginal notes	83
6.19.9	Stock paper	84
6.20	Module <code>dotfill</code>	84
6.21	Module <code>env</code>	85
7	Test	86
7.1	<code>\zref@localaddprop</code>	86
7.2	Module <code>base</code>	86
7.3	Module <code>runs</code>	87
7.4	Module <code>titleref</code>	88
8	Installation	89
8.1	Download	89
8.2	Bundle installation	89
8.3	Package installation	90
8.4	Refresh file name databases	90
8.5	Some details for the interested	90
9	Catalogue	91
10	References	91
11	History	92
	[2006/02/20 v1.0]	92
	[2006/05/03 v1.1]	92
	[2006/05/25 v1.2]	92
	[2006/09/08 v1.3]	92
	[2007/01/23 v1.4]	92
	[2007/02/18 v1.5]	92
	[2007/04/06 v1.6]	93
	[2007/04/17 v1.7]	93
	[2007/04/22 v1.8]	93
	[2007/05/02 v1.9]	93
	[2007/05/06 v2.0]	93
	[2007/05/28 v2.1]	93
	[2008/09/21 v2.2]	93

[2008/10/01 v2.3]	93
[2009/08/07 v2.4]	93
[2009/12/06 v2.5]	93
[2009/12/07 v2.6]	94
[2009/12/08 v2.7]	94
[2010/03/26 v2.8]	94
[2010/03/29 v2.9]	94
[2010/04/08 v2.10]	94
[2010/04/15 v2.11]	94
[2010/04/17 v2.12]	94
[2010/04/19 v2.13]	94
[2010/04/22 v2.14]	95
[2010/04/23 v2.15]	95
[2010/04/28 v2.16]	95
[2010/05/01 v2.17]	95
[2010/05/13 v2.18]	95
[2010/10/22 v2.19]	95
[2011/02/12 v2.20]	96
[2011/03/18 v2.21]	96
[2011/10/05 v2.22]	96
[2011/12/05 v2.23]	96
[2012/04/04 v2.24]	96
[2016/05/16 v2.25]	96
12 Index	96

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_{your key}_key{<some text>}
```

This approach is much more flexible:

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

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

1.3 Interfaces

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

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

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

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

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

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

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

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

2 Interface for programmers

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

2.1 Entities

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

`\Z@R@myref → \default{2.1}\page{7}`

The generic format of a entry is:

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

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

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

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

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

2.2 Property list

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

`\zref@newlist {⟨listname⟩}`

Declares a new empty property list.

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

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

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

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

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

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

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

Executes *⟨then⟩* if the list exists or *⟨else⟩* otherwise.

`\zref@iflistcontainsprop {⟨listname⟩} {⟨propname⟩} {⟨then⟩} {⟨else⟩}`

Executes *⟨then⟩* if the property *⟨propname⟩* is part of property list *⟨listname⟩* or otherwise it runs the *⟨else⟩* part.

2.3 Property

`\zref@newprop* {⟨propname⟩} [⟨default⟩] {⟨value⟩}`

This command declares and configures a new property with name *⟨propname⟩*.

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

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

`\zref@setcurrent {⟨propname⟩} {⟨value⟩}`

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

`\zref@getcurrentexp2 {⟨propname⟩}`

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

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

`\zref@propexists {⟨propname⟩} {⟨then⟩}`

Calls *⟨then⟩* if the property *⟨propname⟩* is available or generates an error message otherwise.

`\zref@ifpropundefinedexp {⟨propname⟩} {⟨then⟩} {⟨else⟩}`

Calls *⟨then⟩* or *⟨else⟩* depending on the existence of property *⟨propname⟩*.

2.4 Reference generation

`\zref@label {⟨refname⟩}`

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

`\zref@labelbylist {⟨refname⟩} {⟨listname⟩}`

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

`\zref@labelbyprops {⟨refname⟩} {⟨propnameA⟩,⟨propnameB⟩,...}`

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

`\zref@newlabel {⟨refname⟩} {...}`

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

2.5 Data extraction

`\zref@extractdefaultexp2 {⟨refname⟩} {⟨propname⟩} {⟨default⟩}`

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

`\zref@extractexp2 {⟨refname⟩} {⟨propname⟩}`

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

Example for page references:

```

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

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

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

`\zref@refused {⟨refname⟩}`

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

`\zref@def@extract {⟨cmd⟩} {⟨refname⟩} {⟨propname⟩}`
`\zref@def@extractdefault {⟨cmd⟩} {⟨refname⟩} {⟨propname⟩} {⟨default⟩}`

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

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

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

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

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

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

Test whether a reference provides a property.

2.6 Setup

`\zref@default`

Holds the global default for unknown values.

`\zref@setdefault {<value>}`

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

`\zref@setmainlist {<value>}`

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

2.7 Declared properties

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

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`:

$$\text{\zref}\{x\} \equiv \text{\zref}[\text{default}]\{x\}$$

`\zpageref {⟨refname⟩}^babel`

Convenience macro, similar to `\pageref`.

$$\text{\zpageref}\{x\} \equiv \text{\zref}[\text{page}]\{x\}$$

`\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. 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}[2016/05/16]
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}[2016/05/16]
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`.

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

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

`title=<value>`

Sets the current title.

`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.

`\zsavepos {<refname>}`

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

`\zsaveposx {<refname>}`

`\zsaveposy {<refname>}`

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

`\zposxexp {<refname>}`

`\zposyexp {<refname>}`

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

Do not rely on absolute page numbers. Because of problems with the origin the numbers may differ in DVI or PDF mode of 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 `dotfill`

`\zdotfill`

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

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

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

`min=<count value>`

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

`unit=<dimen value>`

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

`dot=<value>`

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

3.15 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.

3.16 Module `xr`

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

`\zexternaldocument * [<prefix>]babel {<external document>} [<url>]`

See `\externaldocument` for a description of this option. The found labels also get a property `externaldocument` that remembers `<external document>`. The standard reference scheme and the scheme of this package use different name spaces for reference names. If the external document uses both systems. Then one import statement would put the names in one namespace and probably causing problems with multiple references of the same name. Thus the star form only looks

for `\newlabel` in the `.aux` files, whereas without star only `\zref@newlabels` are used.

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

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

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

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

The following setup options are available:

ext: It sets the default extension.

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

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

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

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

`\zref@xr@ext`

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

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{chaptile}{#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   \zsavepos{firstcol}Hello&\zsavepos{secondcol}World\
158   \zsavepos{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 [2016/05/16 v2.25 A new reference scheme for LaTeX (HO)]%

```

6.1.2 Load basic module

```

191 \RequirePackage{zref-base}[2016/05/16]

```

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}[2016/05/16]%
197     }%
198   }%
199 }
200 \ZREF@temp{abspage}
201 \ZREF@temp{counter}
202 \ZREF@temp{dotfill}
203 \ZREF@temp{hyperref}
204 \ZREF@temp{lastpage}
205 \ZREF@temp{marks}
206 \ZREF@temp{nextpage}
207 \ZREF@temp{pageattr}
208 \ZREF@temp{pagelayout}
209 \ZREF@temp{perpage}
210 \ZREF@temp{runs}
211 \ZREF@temp{savepos}

```

```

212 \ZREF@temp{thepage}
213 \ZREF@temp{titleref}
214 \ZREF@temp{totpages}
215 \ZREF@temp{user}
216 \ZREF@temp{xr}

217 \ProcessOptions\relax
218 \</package>

```

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:

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

\ZREF@: Internal macros.

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

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

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

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

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

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

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

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

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

ZREF@TR: Setup for module *titleref*.

6.2.2 Identification

```

219 <*base>
220 \NeedsTeXFormat{LaTeX2e}
221 \ProvidesPackage{zref-base}%
222   [2016/05/16 v2.25 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     }
```

```

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@wu@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 L^AT_EX undefined references sets a flag and generate a warning. Both actions are not expandable.
- Babel's support for its shorthand uses commands that use non-expandable assignments. However currently there is hope, that primitives are added to pdfL^AT_EX that allows the detection of contexts. Then the shorthand can

detect, if they are executed inside `\csname` and protect themselves automatically.

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

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

`\zifrefundefined` If a reference #1 is undefined, then macro `\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 }
```

`\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 L^AT_EX 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@activestrue\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@gobbletwo
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 \base
```

6.3 Module user

```
959 (*user)
960 \NeedsTeXFormat{LaTeX2e}
961 \ProvidesPackage{zref-user}%
962   [2016/05/16 v2.25 Module user for zref (H0)]%
963 \RequirePackage{zref-base}[2016/05/16]
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 \LaTeX 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
LATEX 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   [2016/05/16 v2.25 Module abspage for zref (HO)]%
1001 \RequirePackage{zref-base}[2016/05/16]
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 </abspage>
```

6.5 Module `counter`

```

1018 <*counter>
1019 \NeedsTeXFormat{LaTeX2e}
1020 \ProvidesPackage{zref-counter}%

```

```

1021 [2016/05/16 v2.25 Module counter for zref (H0)]%
1022 \RequirePackage{zref-base}[2016/05/16]
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 </counter>

```

6.6 Module `lastpage`

```

1038 (*lastpage)
1039 \NeedsTeXFormat{LaTeX2e}
1040 \ProvidesPackage{zref-lastpage}%
1041 [2016/05/16 v2.25 Module lastpage for zref (H0)]%
1042 \RequirePackage{zref-base}[2016/05/16]
1043 \RequirePackage{zref-abspage}[2016/05/16]
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@filesw
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 \lastpage

```

6.7 Module thepage

```

1081 (*thepage)
1082 \NeedsTeXFormat{LaTeX2e}
1083 \ProvidesPackage{zref-thepage}%
1084 [2016/05/16 v2.25 Module thepage for zref (HO)]%
1085 \RequirePackage{zref-base}[2016/05/16]
1086 \ifx\ZREF@base@ok Y%
1087 \else
1088 \expandafter\endinput
1089 \fi
1090 \RequirePackage{atbegshi}[2011/10/05]
1091 \RequirePackage{zref-abspage}[2016/05/16]
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   [2016/05/16 v2.25 Module nextpage for zref (H0)]%
1125 \RequirePackage{zref-base}[2016/05/16]
1126 \ifx\ZREF@base@ok Y%
1127 \else
1128   \expandafter\endinput
1129 \fi

1130 \RequirePackage{zref-abspage}[2016/05/16]
1131 \RequirePackage{zref-thepage}[2016/05/16]
1132 \RequirePackage{zref-lastpage}[2016/05/16]
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@last\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   [2016/05/16 v2.25 Module totpages for zref (HO)]%
1216 \RequirePackage{zref-base}[2016/05/16]
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-abspace}[2016/05/16]
1222 \RequirePackage{zref-lastpage}[2016/05/16]

```

`\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}{abspace}{0}%
1225 }

```

Also we mark the reference `LastPage` as used:

```

1226 \AtBeginDocument{%
1227   \zref@refused{LastPage}%
1228 }
1229 \end{totpages}

```

6.10 Module pagelayout

```

1230 (*pagelayout)
1231 \NeedsTeXFormat{LaTeX2e}
1232 \ProvidesPackage{zref-pagelayout}%
1233   [2016/05/16 v2.25 Module pagelayout for zref (HO)]%
1234 \RequirePackage{zref-base}[2016/05/16]
1235 \ifx\ZREF@base@ok Y%
1236 \else
1237   \expandafter\endinput
1238 \fi
1239 \RequirePackage{zref-thepage}[2016/05/16]
1240 \RequirePackage{ifluatex}[2010/03/01]
1241 \RequirePackage{atveryend}[2010/03/24]

```

6.10.1 Support for LuaTeX

```

1242 \ifluatex
1243   \ifnum\luatexversion<39 %
1244   \else
1245     \begingroup
1246       \escapechar=-1 %
1247       \def\ZREF@temp#1{%
1248         \ltx@ifundefined{\string#1}{%
1249           \let#1\ltx@undefined
1250           \directlua{%
1251             if tex.enableprimitives then %
1252               tex.enableprimitives('', {\string#1'})%
1253             end%
1254           }%
1255         \ltx@ifundefined{\string#1}{%
1256           }{%
1257             \global#1=#1%

```

```

1258         \@PackageInfoNoLine{zref-pagelayout}{%
1259             \string#1 enabled%
1260         }%
1261     }%
1262 }{%
1263 }%
1264 \ZREF@temp\pdfpagewidth
1265 \ZREF@temp\pdfpageheight
1266 \ZREF@temp\pdfhorigin
1267 \ZREF@temp\pdfvorigin
1268 \endgroup
1269 \fi
1270 \fi

```

6.10.2 Define layout properties

```

1271 \def\ZREF@temp#1{%
1272     \begingroup
1273     \escapechar=-1 %
1274     \ltx@ifundefined{\string#1}{\endgroup}{%
1275         \edef\x{%
1276             \endgroup
1277             \noexpand\zref@newprop*{\string#1}%
1278                 [\noexpand\number\noexpand#1]% hash-ok
1279                 {\noexpand\number\noexpand#1}%
1280             \noexpand\zref@addprop{thepage}{\string#1}%
1281         }%
1282         \x
1283     }%
1284 }
1285 \ZREF@temp\mag
1286 \ZREF@temp\paperwidth
1287 \ZREF@temp\paperheight
1288 \ZREF@temp\stockwidth % memoir.cls, crop.sty
1289 \ZREF@temp\stockheight % memoir.cls, crop.sty
1290 \ZREF@temp\mediawidth % VTeX
1291 \ZREF@temp\mediaheight % VTeX
1292 \ZREF@temp\pdfpagewidth
1293 \ZREF@temp\pdfpageheight
1294 \ZREF@temp\pdfhorigin
1295 \ZREF@temp\pdfvorigin
1296 \ZREF@temp\hoffset
1297 \ZREF@temp\voffset
1298 \ZREF@temp\topmargin
1299 \ZREF@temp\oddsidemargin
1300 \ZREF@temp\evensidemargin
1301 \ZREF@temp\textwidth
1302 \ZREF@temp\textheight
1303 \ZREF@temp\headheight
1304 \ZREF@temp\headsep
1305 \ZREF@temp\footskip
1306 \ZREF@temp\marginparwidth
1307 \ZREF@temp\marginparsep
1308 \ZREF@temp\columnwidth
1309 \ZREF@temp\columnsep
1310 \ZREF@temp\trimedge % memoir.cls
1311 \ZREF@temp\spinemargin % memoir.cls
1312 \ZREF@temp\foremargin % memoir.cls
1313 \ZREF@temp\trintop % memoir.cls

```

```

1314 \ZREF@temp\uppermargin % memoir.cls
1315 \ZREF@temp\headmargin % memoir.cls
1316 \zref@newprop*{outputboxwd}[Opt]{\AtBeginShipoutBoxWidth}
1317 \zref@newprop*{outputboxht}[Opt]{\AtBeginShipoutBoxHeight}
1318 \zref@newprop*{outputboxdp}[Opt]{\AtBeginShipoutBoxDepth}
1319 \zref@addprops{thepage}{outputboxwd,outputboxht,outputboxdp}

\ifZREF@pl@list
1320 \ltx@newif\ifZREF@pl@list

\zref@listpagelayout
1321 \ZREF@IfDefinable\zlistpagelayout\def{%
1322   {\global\ZREF@pl@listtrue}%
1323 }

\ZREF@pl@AfterLastShipout
1324 \def\ZREF@pl@AfterLastShipout{%
1325   \ifZREF@pl@list
1326     \edef\ZREF@page@max{\the\value{abspage}}%
1327     \ltx@ifundefined{ZREF@org@testdef}{%
1328       \let\ZREF@org@testdef\@testdef
1329       \def\@testdef##1##2##3{%
1330         \ZREF@org@testdef{##1}{##2}{##3}%
1331         \def\ZREF@temp{##1}%
1332         \ifx\ZREF@temp\ZREF@RefPrefix
1333           \expandafter\gdef\cname##1@##2\endcname{##3}%
1334         \fi
1335       }%
1336     }{%
1337       \AtVeryEndDocument{\ZREF@pl@AtVeryEnd}%
1338     }%
1339 }

\ZREF@pl@AtVeryEnd
1340 \def\ZREF@pl@AtVeryEnd{%
1341   \begingroup
1342   \toks@{Page layout parameters:\MessageBreak}%
1343   \count@=1 %
1344   \ZREF@pl@ListPage
1345   \edef\x{\endgroup
1346     \noexpand\@PackageInfoNoLine{zref-pagelayout}{\the\toks@}%
1347   }%
1348   \x
1349 }

\ZREF@pl@ListPage
1350 \def\ZREF@pl@ListPage{%
1351   \edef\x{%
1352     \toks@={%
1353       \the\toks@
1354       Page \the\count@:\noexpand\MessageBreak
1355       \zref@ifrefundefined{thepage\the\count@}{}{%
1356         \ltx@space\ltx@space mag = %
1357         \zref@extract{thepage\the\count@}{mag}%
1358         \noexpand\MessageBreak
1359         \ZREF@pl@ListEntry{paperwidth}%

```

```

1360     \ZREF@pl@ListEntry{paperheight}%
1361     \ZREF@pl@ListEntry{stockwidth}%
1362     \ZREF@pl@ListEntry{stockheight}%
1363     \ZREF@pl@ListEntry{mediawidth}%
1364     \ZREF@pl@ListEntry{mediaheight}%
1365     \ZREF@pl@ListEntry{pdfpagewidth}%
1366     \ZREF@pl@ListEntry{pdfpageheight}%
1367     \ZREF@pl@ListEntry{pdfhorigin}%
1368     \ZREF@pl@ListEntry{pdfvorigin}%
1369     \ZREF@pl@ListEntry{hoffset}%
1370     \ZREF@pl@ListEntry{voffset}%
1371     \ZREF@pl@ListEntry{topmargin}%
1372     \ZREF@pl@ListEntry{oddsidemargin}%
1373     \ZREF@pl@ListEntry{evensidemargin}%
1374     \ZREF@pl@ListEntry{textwidth}%
1375     \ZREF@pl@ListEntry{textheight}%
1376     \ZREF@pl@ListEntry{headheight}%
1377     \ZREF@pl@ListEntry{headsep}%
1378     \ZREF@pl@ListEntry{footskip}%
1379     \ZREF@pl@ListEntry{marginparwidth}%
1380     \ZREF@pl@ListEntry{marginparsep}%
1381     \ZREF@pl@ListEntry{columnwidth}%
1382     \ZREF@pl@ListEntry{columnsep}%
1383     \ZREF@pl@ListEntry{trimedge}%
1384     \ZREF@pl@ListEntry{spinemargin}%
1385     \ZREF@pl@ListEntry{foremargin}%
1386     \ZREF@pl@ListEntry{trimtop}%
1387     \ZREF@pl@ListEntry{uppermargin}%
1388     \ZREF@pl@ListEntry{headmargin}%
1389   }%
1390 }%
1391 } \x
1392 \ifnum\ZREF@page@max>\count@
1393   \advance\count@ by\ltx@one
1394 \else
1395   \expandafter\ltx@gobble
1396 \fi
1397 \ZREF@pl@ListPage
1398 }

```

\ZREF@pl@ListEntry

```

1399 \def\ZREF@pl@ListEntry#1{%
1400   \zref@ifpropundefined{#1}{%
1401     }{%
1402       \zref@ifrefcontainsprop{thepage\the\count@}{#1}{%
1403         \ltx@space\ltx@space#1 = %
1404         \zref@extract{thepage\the\count@}{#1}sp = %
1405         \the\dimexpr\zref@extract{thepage\the\count@}{#1}sp\relax
1406         \noexpand\MessageBreak
1407       }{%
1408       }%
1409 }
1410 \AfterLastShipout{%
1411   \ZREF@pl@AfterLastShipout
1412 }
1413 </pagelayout>

```

6.11 Module pageattr

```

1414 (*pageattr)
1415 \NeedsTeXFormat{LaTeX2e}
1416 \ProvidesPackage{zref-pageattr}%
1417 [2016/05/16 v2.25 Module pageattr for zref (HO)]%
1418 \RequirePackage{zref-base}[2016/05/16]
1419 \ifx\ZREF@base@ok Y%
1420 \else
1421 \expandafter\endinput
1422 \fi

1423 \RequirePackage{ifluatex}[2010/03/01]

1424 \ifluatex
1425 \ifnum\luatexversion<39 %
1426 \else
1427 \begingroup
1428 \escapechar=-1 %
1429 \def\ZREF@temp#1{%
1430 \ltx@ifundefined{\string#1}{%
1431 \let#1\ltx@undefined
1432 \directlua{%
1433 if tex.enableprimitives then %
1434 tex.enableprimitives('', {\string#1})%
1435 end%
1436 }%
1437 \ltx@ifundefined{\string#1}{%
1438 }{%
1439 \global#1=#1%
1440 \@PackageInfoNoLine{zref-pageattr}{%
1441 \string#1 enabled%
1442 }%
1443 }%
1444 }{%}%
1445 }%
1446 \ZREF@temp\pdfpageattr
1447 \ZREF@temp\pdfpagesattr
1448 \endgroup
1449 \fi
1450 \fi

1451 \let\ZREF@temp=N%
1452 \ltx@ifundefined{pdfpageattr}{%
1453 \@PackageInfoNoLine{zref-pageattr}{%
1454 \string\pdfpageattr\space is not available%
1455 }%
1456 \def\zref@pdfpageattr#1{%
1457 \def\zref@pdfpageattr@used#1{%
1458 }{%
1459 \RequirePackage{zref-thepage}[2016/05/16]%
1460 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfpageattr}}%
1461 \zref@addprop{thepage}{pdfpageattr}%
1462 \let\ZREF@temp=Y%
1463 }
1464 \ltx@ifundefined{pdfpagesattr}{%
1465 \@PackageInfoNoLine{zref-pageattr}{%
1466 \string\pdfpagesattr\space is not available%
1467 }%
1468 \def\zref@pdfpagesattr{%
1469 \def\zref@pdfpagesattr@used{%

```

```

1470 }{%
1471   \RequirePackage{zref-lastpage}[2016/05/16]%
1472   \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfpagesattr}}%
1473   \zref@addprop{LastPage}{pdfpagesattr}%
1474   \let\ZREF@temp=Y%
1475 }%

1476 \ifx\ZREF@temp N%
1477   \expandafter\endinput
1478 \fi

1479 \RequirePackage{zref-abspage}[2016/05/16]
1480 \RequirePackage{atveryend}[2010/03/24]
1481 \RequirePackage{pdftexcmds}[2010/04/01]
1482 \let\ZREF@temp=Y%
1483 \ltx@ifundefined{pdf@escapehex}{\let\ZREF@temp=N}{%}
1484 \ltx@ifundefined{pdf@unescapehex}{\let\ZREF@temp=N}{%}
1485 \ifx\ZREF@temp N%
1486   \let\zref@hex\ltx@firstofone
1487   \let\zref@unhex\ltx@firstofone
1488 \else
1489   \let\zref@hex\pdf@escapehex
1490   \let\zref@unhex\pdf@unescapehex
1491 \fi

\ifZREF@pa@list

1492 \ltx@newif\ifZREF@pa@list

\zref@listpageattr

1493 \ZREF@ifdefinable\zlistpageattr\def{%
1494   {\ZREF@pa@listtrue}}%
1495 }

\ZREF@pa@AfterLastShipout

1496 \def\ZREF@pa@AfterLastShipout{%
1497   \ifZREF@pa@list
1498     \edef\ZREF@page@max{\the\value{abspage}}%
1499     \ltx@ifundefined{ZREF@org@testdef}{%
1500       \let\ZREF@org@testdef\@testdef
1501       \def\@testdef##1##2##3{%
1502         \ZREF@org@testdef{##1}{##2}{##3}%
1503         \def\ZREF@temp{##1}%
1504         \ifx\ZREF@temp\ZREF@RefPrefix
1505           \expandafter\xdef\csname##1@##2\endcsname{##3}%
1506         \fi
1507       }%
1508     }{%
1509       \AtVeryEndDocument{\ZREF@pa@AtVeryEnd}%
1510     }%
1511 }

\ZREF@pa@AtVeryEnd

1512 \ltx@ifundefined{pdfpageattr}{%
1513   \def\ZREF@pa@AtVeryEnd{%
1514   }{%
1515     \def\ZREF@pa@AtVeryEnd{%
1516       \begingroup
1517       \toks@{\List of \ltx@backslashchar pdfpageattr:\MessageBreak}%
1518       \count@=1 %

```

```

1519 \ZREF@pa@ListPage
1520 \edef\x{\endgroup
1521 \noexpand\@PackageInfoNoLine{zref-pageattr}{%
1522 \the\toks@
1523 }%
1524 }%
1525 \x
1526 }%

```

\zref@pageattr

```

1527 \def\zref@pageattr#1{%
1528 \zref@unhex{%
1529 \zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}%
1530 }%
1531 }

```

\zref@pageattr@used

```

1532 \ZREF@Robust\def\zref@pageattr@used#1{%
1533 \zref@refused{thepage\ZREF@number{#1}}%
1534 }

```

\ZREF@pa@ListPage

```

1535 \def\ZREF@pa@ListPage{%
1536 \edef\x{%
1537 \toks@={%
1538 \the\toks@
1539 Page \the\count@:%
1540 \noexpand\MessageBreak
1541 \zref@ifrefundefined{thepage\the\count@}{-}{%
1542 <<\zref@pdfpageattr\count@>>%
1543 \noexpand\MessageBreak
1544 }%
1545 }%
1546 }\x
1547 \ifnum\ZREF@page@max>\count@
1548 \advance\count@ by\ltx@one
1549 \else
1550 \expandafter\ltx@gobble
1551 \fi
1552 \ZREF@pa@ListPage
1553 }%
1554 }

```

```

1555 \ltx@ifundefined{pdfpagesattr}{-}{%
1556 }{-}%

```

\zref@pdfpagesattr

```

1557 \def\zref@pdfpagesattr{%
1558 \zref@unhex{%
1559 \zref@extract{LastPage}{pdfpagesattr}%
1560 }%
1561 }%

```

\zref@pdfpagesattr@used

```

1562 \ZREF@Robust\def\zref@pdfpagesattr@used{%
1563 \zref@refused{LastPage}%
1564 }%

```

```

1565 \ltx@LocalAppendToMacro\ZREF@pa@AtVeryEnd{%
1566 \@PackageInfoNoLine{zref-pageattr}{%
1567 \ltx@backslashchar pdfpagesattr:\MessageBreak
1568 <<\zref@pdfpagesattr>>%
1569 \MessageBreak
1570 }%
1571 }%
1572 }
1573 \AfterLastShipout{%
1574 \ZREF@pa@AfterLastShipout
1575 }
1576 />pageattr)

```

6.12 Module marks

```

1577 (*marks)
1578 \NeedsTeXFormat{LaTeX2e}
1579 \ProvidesPackage{zref-marks}%
1580 [2016/05/16 v2.25 Module marks for zref (H0)]%
1581 \RequirePackage{zref-base}[2016/05/16]
1582 \ifx\ZREF@base@ok Y%
1583 \else
1584 \expandafter\endinput
1585 \fi

1586 \newcommand*{\zref@marks@register}[3][{}]{%
1587 \edef\ZREF@TempName{#1}%
1588 \edef\ZREF@TempNum{\ZREF@number{#2}}%
1589 \ifnum\ZREF@TempNum<\ltx@zero %
1590 \PackageError\ZREF@name{%
1591 \string\zref@marks@register\ltx@space is called with invalid%
1592 \MessageBreak
1593 marks register number (\ZREF@TempNum)%
1594 }{%
1595 Use ‘0’ or the command, defined by \string\newmarks.\MessageBreak
1596 \@ehc
1597 }%
1598 \else
1599 \ifx\ZREF@TempName\ltx@empty
1600 \edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}%
1601 \else
1602 \edef\ZREF@TempName{marks\ZREF@TempName}%
1603 \fi
1604 \ZREF@MARKS@DefineProp{top}%
1605 \ZREF@MARKS@DefineProp{first}%
1606 \ZREF@MARKS@DefineProp{bot}%
1607 \kv@parse{#3}{%
1608 \ifx\kv@value\relax
1609 \def\kv@value{top,first,bot}%
1610 \fi
1611 \edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}%
1612 \ifx\ZREF@temp\ZREF@STAR
1613 \edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}%
1614 \zref@newlist\kv@key
1615 \fi
1616 \expandafter\comma@parse\expandafter{\kv@value}{%
1617 \ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else
1618 \ifx\comma@entry\ZREF@NAME@first 1\else

```

```

1619         \ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space
1620         \PackageWarning{zref-marks}{%
1621         Use 'top', 'first' or 'bot' for the list values%
1622         \MessageBreak
1623         in the third argument of \string\zref@marks@register.%
1624         \MessageBreak
1625         Ignoring unkown value '\comma@entry'%
1626         }%
1627     \else
1628         \zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}%
1629     \fi
1630     \ltx@gobble
1631 }%
1632 \ltx@gobbletwo
1633 }%
1634 \fi
1635 }
1636 \def\ZREF@STAR{*}
1637 \def\ZREF@NAME@top{top}
1638 \def\ZREF@NAME@first{first}
1639 \def\ZREF@NAME@bot{bot}
1640 \def\ZREF@MARKS@DefineProp#1{%
1641     \zref@ifpropundefined{#1\ZREF@TempName}{%
1642         \ifnum\ZREF@TempNum=\ltx@zero
1643             \begingroup
1644             \edef\x{\endgroup
1645                 \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
1646                     \expandafter\noexpand\csname#1mark\endcsname
1647                 }%
1648             }%
1649             \x
1650         \else
1651             \begingroup
1652             \edef\x{\endgroup
1653                 \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
1654                     \expandafter\noexpand\csname#1marks\endcsname
1655                     \ZREF@TempNum
1656                 }%
1657             }%
1658             \x
1659         \fi
1660     }{%
1661         \PackageWarning{zref-marks}{%
1662         \string\zref@marks@register\ltx@space does not generate the%
1663         \MessageBreak
1664         new property '#1\ZREF@TempName', because\MessageBreak
1665         it is already defined%
1666         }%
1667     }%
1668 }
1669 </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.

```

1670 <*runs>
1671 \NeedsTeXFormat{LaTeX2e}

```

```

1672 \ProvidesPackage{zref-runs}%
1673 [2016/05/16 v2.25 Module runs for zref (H0)]%

\zruns

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

1687 \</runs>

```

6.14 Module perpage

```

1688 (*perpage)
1689 \NeedsTeXFormat{LaTeX2e}
1690 \ProvidesPackage{zref-perpage}%
1691 [2016/05/16 v2.25 Module perpage for zref (H0)]%
1692 \RequirePackage{zref-base}[2016/05/16]
1693 \ifx\ZREF@base@ok Y%
1694 \else
1695   \expandafter\endinput
1696 \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.

```
1697 \RequirePackage{zref-abspage}[2016/05/16]
```

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

```

1698 \ZREF@NewPropPageValue
1699 \zref@newlist{perpage}
1700 \zref@addprops{perpage}{abspage,page,pagevalue}

```

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

```
1701 \newcounter{zpage}
```

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

```
1702 \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`.

```

1703 \let\ZREF@org@stepcounter\stepcounter
1704 \def\stepcounter#1{%
1705   \ifcsname @stepcounterhook@#1\endcsname
1706     \csname @stepcounterhook@#1\endcsname
1707   \fi
1708   \ZREF@org@stepcounter{#1}%
1709 }

```

`\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.

```

1710 \ZREF@ifDefinable\zmakeperpage\def{%
1711   {%
1712     \@ifnextchar[\ZREF@makeperpage@opt{\ZREF@@makeperpage[\ltx@zero]}%
1713   }%
1714 }

```

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

```

1715 \def\ZREF@makeperpage@opt[#1]{%
1716   \begingroup
1717   \edef\x{\endgroup
1718     \noexpand\ZREF@@makeperpage[\number\numexpr#1-1\relax]%
1719   }%
1720   \x
1721 }

1722 \def\ZREF@@makeperpage[#1]#2{%
1723   \@ifundefined{stepcounterhook@#2}{%
1724     \expandafter\gdef\csname stepcounterhook@#2\endcsname{%
1725     }%
1726     \expandafter\gdef\csname ZREF@perpage@#2\endcsname{%
1727       \ZREF@@perpage@step{#2}{#1}%
1728     }%
1729     \expandafter\g@addto@macro\csname stepcounterhook@#2\endcsname{%
1730       \ifcsname ZREF@perpage@#2\endcsname
1731         \csname ZREF@perpage@#2\endcsname
1732       \fi
1733     }%
1734 }

```

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

```

1735 \def\ZREF@@perpage@step#1#2{%

```

First the reference is generated.

```

1736   \global\advance\c@zref@unique\ltx@one
1737   \begingroup
1738   \expandafter
1739   \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.

```

1740   \zref@ifrefundefined\thezref@unique{%
1741     \global\c@zpage=\c@page
1742     \global\let\thezpage\thepage
1743     \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1744       \number\c@abspage
1745     }%
1746   }{%

```

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

```

1747     \global\c@zpage=\zref@extract\thezref@unique{pagevalue}\relax
1748     \xdef\thezpage{\noexpand\zref@extract{\thezref@unique}{page}}%
1749     \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%

```

```

1750      \zref@extractdefault\thezref@unique
1751      {abspage}{\number\c@abspage}%
1752    }%
1753  }%

```

Page changes are detected by a changed absolute page number.

```

1754    \expandafter\ifx\csname ZREF@abspage@#1\expandafter\endcsname
1755      \csname ZREF@currentabspage@#1\endcsname
1756    \else
1757      \global\csname c@#1\endcsname=#2\relax
1758      \global\expandafter\let
1759        \csname ZREF@currentabspage@#1\expandafter\endcsname
1760        \csname ZREF@abspage@#1\endcsname
1761    \fi
1762  \endgroup
1763 }

```

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

```

1764 \ZREF@ifDefinable\zunmakeperpage\def{%
1765   #1{%
1766     \global\expandafter
1767     \let\csname ZREF@perpage@#1\endcsname\@undefined
1768   }%
1769 }

```

1770 `/perpage`

6.15 Module `titleref`

```

1771 (*titleref)
1772 \NeedsTeXFormat{LaTeX2e}
1773 \ProvidesPackage{zref-titleref}%
1774   [2016/05/16 v2.25 Module titleref for zref (H0)]%
1775 \RequirePackage{zref-base}[2016/05/16]
1776 \ifx\ZREF@base@ok Y%
1777 \else
1778   \expandafter\endinput
1779 \fi
1780 \RequirePackage{getttitlestring}[2009/12/08]

```

6.15.1 Implementation

```
1781 \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.

```

1782 \ZREF@NewPropTitle
1783 \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`

```

1784 \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 re-
moved. The hook is executed before the expansion of \zref@titleref@current.
1785 \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 accidently.
1786 \ZREF@Robust\def\zref@titleref@cleanup#1{%
1787   \begingroup
1788   \toks@\expandafter{%
1789     \ZREF@titleref@hook
1790     #1%
1791   }%
1792   \expandafter\endgroup
1793   \expandafter\def\expandafter\ZREF@titleref@hook\expandafter{%
1794     \the\toks@
1795   }%
1796 }%

\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.
1797 \newif\ifzref@titleref@stripperperiod
1798 \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.
1799 \ZREF@Robust\def\zref@titleref@setcurrent#1{%
1800   \ifzref@titleref@expand
1801     \GetTitleStringExpand{#1}%
1802   \else
1803     \GetTitleStringNonExpand{#1}%
1804   \fi
1805   \edef\zref@titleref@current{%
1806     \detokenize\expandafter{\GetTitleStringResult}%
1807   }%
1808   \ifzref@titleref@stripperperiod
1809     \edef\zref@titleref@current{%
1810       \expandafter\ZREF@stripperperiod\zref@titleref@current
1811       \ltx@empty.\ltx@empty\@nil
1812     }%
1813   \fi
1814 }%
1815 \GetTitleStringDisableCommands{%
1816   \ZREF@titleref@hook
1817 }

\ZREF@stripperperiod If \ZREF@stripperperiod is called, the argument consists of space tokens and tokens
with catcode 12 (other), because of  $\epsilon$ -TeX's \detokenize.
1818 \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`.

```
1819 \define@key{ZREF@TR}{expand}[true]{%
1820   \csname zref@titleref@expand#1\endcsname
1821 }%
1822 \define@key{ZREF@TR}{stripperperiod}[true]{%
1823   \csname zref@titleref@stripperperiod#1\endcsname
1824 }%
1825 \define@key{ZREF@TR}{cleanup}{%
1826   \zref@titleref@cleanup{#1}%
1827 }%
1828 \define@key{ZREF@TR}{title}{%
1829   \def\zref@titleref@current{#1}%
1830 }%
1831 \ZREF@ifDefinable\ztitlerefsetup\def{%
1832   {\kvsetkeys{ZREF@TR}}}%
1833 }%
```

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

```
1834 \ZREF@ifDefinable\ztitleref\def{%
1835   {\zref@wrapper@babel\ZREF@titleref}%
1836 }%
1837 \def\ZREF@titleref#1{%
1838   \begingroup
1839     \zref@refused{#1}%
1840     \let\label\ltx@gobble
1841     \zref@extract{#1}{title}%
1842   \endgroup
1843 }%
```

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.

```
1844 \AtBeginDocument{%
1845   \ZREF@patch{@caption}{%
1846     \long\def@caption#1[#2]{%
1847       \zref@titleref@setcurrent{#2}%
1848       \ZREF@org@@caption{#1}[{#2}]%
1849     }%
1850   }%
```

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

```
1851   \ZREF@patch{@part}{%
1852     \def@part[#1]{%
1853       \zref@titleref@setcurrent{#1}%
1854       \ZREF@org@@part[{#1}]%
1855     }%
1856   }%
1857   \ZREF@patch{@chapter}{%
1858     \def@chapter[#1]{%
1859       \zref@titleref@setcurrent{#1}%
1860       \ZREF@org@@chapter[{#1}]%
1861     }%
1862   }%
```

```

1861 }%
1862 }%
1863 \ZREF@patch{@sect}{%
1864   \def\@sect#1#2#3#4#5#6[#7]{%
1865     \zref@titleref@setcurrent{#7}%
1866     \ZREF@org@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]%
1867   }%
1868 }%

```

The star versions of the section commands.

```

1869 \ZREF@patch{@spart}{%
1870   \def\@spart#1{%
1871     \zref@titleref@setcurrent{#1}%
1872     \ZREF@org@@spart{#1}%
1873   }%
1874 }%
1875 \ZREF@patch{@schapter}{%
1876   \def\@schapter#1{%
1877     \zref@titleref@setcurrent{#1}%
1878     \ZREF@org@@schapter{#1}%
1879   }%
1880 }%
1881 \ZREF@patch{@ssect}{%
1882   \def\@ssect#1#2#3#4#5{%
1883     \zref@titleref@setcurrent{#5}%
1884     \ZREF@org@ssect{#1}{#2}{#3}{#4}{#5}%
1885   }%
1886 }%

```

6.15.4 Environment description

```

1887 \ZREF@patch{descriptionlabel}{%
1888   \def\descriptionlabel#1{%
1889     \zref@titleref@setcurrent{#1}%
1890     \ZREF@org@descriptionlabel{#1}%
1891   }%
1892 }%

```

6.15.5 Class memoir

```

1893 \@ifclassloaded{memoir}{%
1894   \ltx@ifundefined{ifheadnameref}{}{%
1895     \def\@chapter[#1]#2{%
1896       \ltx@ifundefined{ch@pt@c}{%
1897         \zref@titleref@setcurrent{#1}%
1898       }{%
1899         \ifx\ch@pt@c\ltx@empty
1900           \zref@titleref@setcurrent{#2}%
1901         \else
1902           \def\NR@temp{#1}%
1903           \ifx\NR@temp\ltx@empty
1904             \expandafter\zref@titleref@setcurrent
1905             \expandafter{\ch@pt@c}%
1906           \else
1907             \ifheadnameref
1908               \zref@titleref@setcurrent{#1}%
1909             \else
1910               \expandafter\zref@titleref@setcurrent
1911               \expandafter{\ch@pt@c}%
1912             \fi

```

```

1913         \fi
1914     \fi
1915 }%
1916 \ZREF@org@@chapter[{#1}]{#2}%
1917 }%
1918 \ZREF@patch{M@sect}{%
1919     \def\M@sect#1#2#3#4#5#6[#7][#8]{%
1920         \ifheadnameref
1921             \zref@titleref@setcurrent{#8}%
1922         \else
1923             \zref@titleref@setcurrent{#7}%
1924         \fi
1925         \ZREF@org@M@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}][{#8}]%
1926     }%
1927 }%
1928 }%
1929 }{}%

```

6.15.6 Class beamer

```

1930 \@ifclassloaded{beamer}{%
1931     \ZREF@patch{beamer@section}{%
1932         \long\def\beamer@section[#1]{%
1933             \zref@titleref@setcurrent{#1}%
1934             \ZREF@org@beamer@section[{#1}]%
1935         }%
1936     }%
1937     \ZREF@patch{beamer@subsection}{%
1938         \long\def\beamer@subsection[#1]{%
1939             \zref@titleref@setcurrent{#1}%
1940             \ZREF@org@beamer@subsection[{#1}]%
1941         }%
1942     }%
1943     \ZREF@patch{beamer@subsubsection}{%
1944         \long\def\beamer@subsubsection[#1]{%
1945             \zref@titleref@setcurrent{#1}%
1946             \ZREF@org@beamer@subsubsection[{#1}]%
1947         }%
1948     }%
1949 }{}%

```

6.15.7 Package titlesec

```

1950 \@ifpackageloaded{titlesec}{%
1951     \ZREF@patch{ttl@sect@i}{%
1952         \def\ttl@sect@i#1#2[#3]#4{%
1953             \zref@titleref@setcurrent{#4}%
1954             \ZREF@org@ttl@sect@i{#1}{#2}[{#3}]{#4}%
1955         }%
1956     }%
1957     \ZREF@patch{ttl@straight@i}{%
1958         \def\ttl@straight@i#1[#2]#3{%
1959             \def\ZREF@temp{#2}%
1960             \ifx\ZREF@temp\ltx@empty
1961                 \zref@titleref@setcurrent{#3}%
1962             \else
1963                 \zref@titleref@setcurrent{#2}%
1964             \fi
1965             \ZREF@org@ttl@straight@i{#1}[{#2}]{#3}%
1966         }%

```

```

1967 }%
1968 }{}%

```

6.15.8 Package longtable

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

```

1969 \@ifpackageloaded{longtable}{%
1970   \ZREF@patch{LT@c@ption}{%
1971     \def\LT@c@ption#1[#2]#3{%
1972       \ZREF@org@LT@c@ption{#1}[{#2}]{#3}%
1973       \zref@titleref@setcurrent{#2}%
1974     }%
1975   }%
1976 }{}%

```

6.15.9 Package listings

Package listings: support for its caption.

```

1977 \@ifpackageloaded{listings}{%
1978   \ZREF@patch{lst@MakeCaption}{%
1979     \def\lst@MakeCaption{%
1980       \ifx\lst@label\ltx@empty
1981       \else
1982         \expandafter\zref@titleref@setcurrent\expandafter{%
1983           \lst@@caption
1984         }%
1985       \fi
1986       \ZREF@org@lst@MakeCaption
1987     }%
1988   }%
1989 }{}%

```

6.15.10 Theorems

```

1990 \ZREF@patch{@opargbegintheorem}{%
1991   \def\@opargbegintheorem#1#2#3{%
1992     \zref@titleref@setcurrent{#3}%
1993     \ZREF@org@@opargbegintheorem{#1}{#2}{#3}%
1994   }%
1995 }%
1996 \@ifpackageloaded{amsthm}{%
1997   \begingroup
1998   \edef\x{macro:\string#1\string#2[\string#3]}%
1999   \@onelevel@sanitize\x
2000   \def\y#1->#2\@nil{#1}%
2001   \edef\z{\expandafter\y\meaning\@begintheorem->\@nil}%
2002   \@onelevel@sanitize\z
2003   \expandafter\endgroup
2004   \ifx\x\z
2005     \ZREF@patch{@begintheorem}{%
2006       \def\@begintheorem#1#2[#3]{%
2007         \zref@titleref@setcurrent{#3}%
2008         \ZREF@org@@begintheorem{#1}{#2}[{#3}]%
2009       }%
2010     }%
2011   \fi
2012 }{}%

```

```

2013 }
2014 
```

6.16 Module xr

```

2015 (*xr)
2016 \NeedsTeXFormat{LaTeX2e}
2017 \ProvidesPackage{zref-xr}%
2018 [2016/05/16 v2.25 Module xr for zref (HO)]%
2019 \RequirePackage{zref-base}[2016/05/16]
2020 \ifx\ZREF@base@ok Y%
2021 \else
2022 \expandafter\endinput
2023 \fi

2024 \RequirePackage{keyval}
2025 \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.

```

2026 \zref@newprop{url}{}%
2027 \zref@newprop{urluse}{}%
2028 \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.

```

2029 \providecommand*{\zref@xr@ext}{}%
2030 \ltx@ifundefined{XR@ext}{pdf}{\XR@ext}%
2031 }%

```

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

```

2032 \newif\ifZREF@xr@zreflabel

2033 \SetupKeyvalOptions{%
2034   family=ZREF@XR,%
2035   prefix=ZREF@xr@%
2036 }
2037 \DeclareBoolOption[true]{tozreflabel}
2038 \DeclareBoolOption[false]{toltxlabel}
2039 \DeclareBoolOption{verbose}
2040 \define@key{ZREF@XR}{ext}{}%
2041 \def\zref@xr@{#1}%
2042 }
2043 \DeclareBoolOption[false]{urluse}

```

`\zxrsetup`

```

2044 \newcommand*{\zxrsetup}{}%
2045 \kvsetkeys{ZREF@XR}%
2046 }%

```

`\ZREF@xr@URL`

```

2047 \newcount\ZREF@xr@URL
2048 \ZREF@xr@URL=\ltx@zero

```

`\ZREF@xr@AddURL`

```

2049 \def\ZREF@xr@AddURL#1{%
2050   \begingroup
2051     \def\ZREF@temp{#1}%
2052     \count@=\ltx@one
2053     \ZREF@xr@@AddUrl
2054   \endgroup
2055 }

```

`\ZREF@xr@@AddUrl`

```

2056 \def\ZREF@xr@@AddUrl{%
2057   \ifnum\count@>\ZREF@xr@URL
2058     \global\advance\ZREF@xr@URL by\ltx@one
2059     \xdef\ZREF@xr@theURL{\romannumeral\ZREF@xr@URL}%
2060     \global\expandafter\let
2061       \csname Z@U@\ZREF@xr@theURL\endcsname\ZREF@temp
2062     \@PackageInfo{zref-xr}{%
2063       \ltx@backslashchar Z@U@\ZREF@xr@theURL:\MessageBreak
2064       \ZREF@temp\MessageBreak
2065     }%
2066   \else
2067     \expandafter
2068     \ifx\csname Z@U@\romannumeral\count@\endcsname\ZREF@temp
2069       \xdef\ZREF@xr@theURL{\romannumeral\count@}%
2070     \else
2071       \expandafter\expandafter\expandafter\ZREF@xr@@AddUrl
2072     \fi
2073   \fi
2074 }

```

`\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).

```

2075 \ZREF@ifDefinable\zexternaldocument\def{%
2076   {%
2077     \ZREF@NewPropAnchor
2078     \ZREF@NewPropTitle
2079     \begingroup
2080       \csname @safe@actives@true\endcsname
2081       \makeatletter
2082       \@ifstar{%
2083         \ZREF@xr@zreflabelfalse
2084         \@testopt\ZREF@xr@externaldocument{}%
2085       }{%
2086         \ZREF@xr@zreflabeltrue
2087         \@testopt\ZREF@xr@externaldocument{}%
2088       }%
2089     }%
2090   }%

```

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.

```

2091 \def\ZREF@xr@externaldocument[#1]#2{%

```

```

2092 \def\ZREF@xr@prefix{#1}%
2093 \let\ZREF@xr@filelist\ltx@empty
2094 \edef\ZREF@xr@externalfile{#2}%
2095 \edef\ZREF@xr@file{\ZREF@xr@externalfile.aux}%
2096 \filename@parse{#2}%
2097 \@testopt\ZREF@xr@graburl{#2.\zref@xr@ext}%
2098 }%
2099 \def\ZREF@xr@graburl[#1]{%
2100 \edef\ZREF@xr@url{#1}%
2101 \ifZREF@xr@urluse
2102 \expandafter\ZREF@xr@AddURL\expandafter{\ZREF@xr@url}%
2103 \expandafter\def\expandafter\ZREF@xr@url
2104 \expandafter{\csname Z@U@\ZREF@xr@theURL\endcsname}%
2105 \fi
2106 \ZREF@xr@checkfile
2107 \endgroup
2108 }%

```

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

```

2109 \def\ZREF@xr@checkfile{%
2110 \openin\@inputcheck\ZREF@xr@file\relax
2111 \ifeof\@inputcheck
2112 \PackageWarning{zref-xr}{%
2113 File '\ZREF@xr@file' not found or empty,\MessageBreak
2114 labels not imported%
2115 }%
2116 \else
2117 \PackageInfo{zref-xr}{%
2118 Label \ifZREF@xr@zreflabel (zref) \fi
2119 import from '\ZREF@xr@file'%
2120 }%
2121 \def\ZREF@xr@found{0}%
2122 \def\ZREF@xr@ignored@empty{0}%
2123 \def\ZREF@xr@ignored@zref{0}%
2124 \def\ZREF@xr@ignored@ltx{0}%
2125 \ZREF@xr@processfile
2126 \closein\@inputcheck
2127 \begingroup
2128 \let\on@line\ltx@empty
2129 \PackageInfo{zref-xr}{%
2130 Statistics for '\ZREF@xr@file':\MessageBreak
2131 \ZREF@xr@found\space
2132 \ifZREF@xr@zreflabel zref\else LaTeX\fi\space
2133 label(s) found%
2134 \ifnum\ZREF@xr@ignored@empty>0 %
2135 ,\MessageBreak
2136 \ZREF@xr@ignored@empty\space empty label(s) ignored%
2137 \fi
2138 \ifnum\ZREF@xr@ignored@zref>0 %
2139 ,\MessageBreak
2140 \ZREF@xr@ignored@zref\space
2141 duplicated zref label(s) ignored%
2142 \fi
2143 \ifnum\ZREF@xr@ignored@ltx>0 %
2144 ,\MessageBreak
2145 \ZREF@xr@ignored@ltx\space
2146 duplicated latex label(s) ignored%

```

```

2147     \fi
2148   }%
2149   \endgroup
2150 \fi
2151 \ifx\ZREF@xr@filelist\ltx@empty
2152 \else
2153   \edef\ZREF@xr@file{%
2154     \expandafter\ltx@car\ZREF@xr@filelist\@nil
2155   }%
2156   \edef\ZREF@xr@filelist{%
2157     \expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil
2158   }%
2159   \expandafter\ZREF@xr@checkfile
2160 \fi
2161 }%

```

\ZREF@xr@processfile

```

2162 \def\ZREF@xr@processfile{%
2163   \read\@inputcheck to\ZREF@xr@line
2164   \expandafter\ZREF@xr@processline\ZREF@xr@line..\ZREF@nil
2165   \ifeof\@inputcheck
2166   \else
2167     \expandafter\ZREF@xr@processfile
2168   \fi
2169 }%

```

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

```

2170 \long\def\ZREF@xr@processline#1#2#3\ZREF@nil{%
2171   \def\x{#1}%
2172   \toks@{#2}%
2173   \ifZREF@xr@zreflabel
2174     \ifx\x\ZREF@xr@zref@newlabel
2175       \expandafter
2176       \ZREF@xr@process@zreflabel\ZREF@xr@line...\ZREF@nil
2177     \fi
2178   \else
2179     \ifx\x\ZREF@xr@newlabel
2180       \expandafter
2181       \ZREF@xr@process@label\ZREF@xr@line...[]\ZREF@nil
2182     \fi
2183   \fi
2184   \ifx\x\ZREF@xr@input
2185     \edef\ZREF@xr@filelist{%
2186       \etex@unexpanded\expandafter{\ZREF@xr@filelist}%
2187       {\filename@area\the\toks@}%
2188     }%
2189   \fi
2190 }%
2191 \def\ZREF@xr@process@zreflabel\zref@newlabel#1#2#3\ZREF@nil{%
2192   \edef\ZREF@xr@refname{Z@R@\ZREF@xr@prefix#1}%
2193   \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2194   \def\x{#2}%
2195   \edef\ZREF@xr@tempname{$temp$}%
2196   \edef\ZREF@xr@temprefname{Z@R@\ZREF@xr@tempname}%
2197   \let\ZREF@xr@list\x
2198   \ifx\ZREF@xr@list\ltx@empty
2199     \PackageWarningNoLine{zref-xr}{%
2200       Label ‘#1’ without properties ignored\MessageBreak

```

```

2201     in file '\ZREF@xr@file'%
2202 }%
2203 \edef\ZREF@xr@ignored@empty{%
2204     \the\numexpr\ZREF@xr@ignored@empty+1\relax
2205 }%
2206 \else
2207     \expandafter\ZREF@xr@checklist\x\ZREF@nil
2208     \expandafter\let\csname\ZREF@xr@temprefname\endcsname\x
2209     \expandafter\ltx@LocalAppendToMacro
2210     \csname\ZREF@xr@temprefname\expandafter\endcsname
2211     \expandafter{%
2212         \expandafter\externaldocument\expandafter{%
2213             \ZREF@xr@externalfile
2214         }%
2215     }%
2216     \ZREF@xr@urlcheck\ZREF@xr@tempname
2217     \ifZREF@xr@tozreflabel
2218         \ifundefined{\ZREF@xr@refname}{%
2219             \ifZREF@xr@verbose
2220                 \PackageInfo{zref-xr}{%
2221                     Import to zref label '\ZREF@xr@tempname#1'%
2222                 }%
2223             \fi
2224             \global\expandafter
2225             \let\csname\ZREF@xr@refname\expandafter\endcsname
2226             \csname\ZREF@xr@temprefname\endcsname
2227         }{%
2228             \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2229         }%
2230     \fi
2231     \ifZREF@xr@toltxlabel
2232         \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2233     \fi
2234 \fi
2235 }%
2236 \def\ZREF@xr@process@label\newlabel#1#2#3[#4]#5\ZREF@nil{%
2237     \def\ZREF@xr@refname{Z@R@\ZREF@xr@prefix#1}%
2238     \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2239     \def\x{#2}%
2240     \edef\ZREF@xr@tempname{$temp$}%
2241     \edef\ZREF@xr@temprefname{Z@R@\ZREF@xr@tempname}%
2242     \expandafter\ZREF@xr@scanparams
2243     \csname\ZREF@xr@temprefname\expandafter\endcsname
2244     \x{}{}{}{}{}\ZREF@nil
2245     \ifx\\#4\\%
2246     \else
2247         % ntheorem knows an optional argument at the end of \newlabel
2248         \ZREF@NewPropTheotype
2249         \expandafter\ltx@LocalAppendToMacro
2250         \csname\ZREF@xr@temprefname\endcsname{\theotype{#4}}%
2251     \fi
2252     \expandafter\ltx@LocalAppendToMacro
2253     \csname\ZREF@xr@temprefname\expandafter\endcsname\expandafter{%
2254         \expandafter\externaldocument\expandafter{%
2255             \ZREF@xr@externalfile
2256         }%
2257     }%
2258     \ZREF@xr@urlcheck\ZREF@xr@tempname

```

```

2259 \ifZREF@xr@tozreflabel
2260   \@ifundefined{\ZREF@xr@refname}{%
2261     \ifZREF@xr@verbose
2262       \PackageInfo{zref-xr}{%
2263         Import to zref label '\ZREF@xr@prefix#1'%
2264       }%
2265     \fi
2266     \global\expandafter
2267     \let\csname\ZREF@xr@refname\expandafter\endcsname
2268     \csname\ZREF@xr@temprefname\endcsname
2269   }{%
2270     \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2271   }%
2272 \fi
2273 \ifZREF@xr@toltxlabel
2274   \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2275 \fi
2276 }
2277 \def\ZREF@xr@zref@newlabel{\zref@newlabel}%
2278 \def\ZREF@xr@newlabel{\newlabel}%
2279 \def\ZREF@xr@@input{\@input}%
2280 \def\ZREF@xr@relax{\relax}%

```

\ZREF@xr@tolabel

```

2281 \def\ZREF@xr@tolabel#1#2{%
2282   \ifZREF@xr@verbose
2283     \PackageInfo{zref-xr}{%
2284       Import to LaTeX label '#2'%
2285     }%
2286   \fi
2287   \zref@wrapper@unexpanded{%
2288     \expandafter\xdef\csname r@#2\endcsname{%
2289       {%
2290         \ltx@ifundefined{M@TitleReference}{%
2291           \ltx@ifundefined{TR@TitleReference}{%
2292             \zref@extractdefault{#1}{default}{}}%
2293           {%
2294             \noexpand\TR@TitleReference
2295             {\zref@extractdefault{#1}{default}{}}%
2296             {\zref@extractdefault{#1}{title}{}}%
2297           }%
2298         }{%
2299           \noexpand\M@TitleReference
2300           {\zref@extractdefault{#1}{default}{}}%
2301           {\zref@extractdefault{#1}{title}{}}%
2302         }%
2303       }%
2304     {\zref@extractdefault{#1}{page}{}}%
2305     \ltx@ifpackageloaded{nameref}{%
2306       {\zref@extractdefault{#1}{title}{}}%
2307       {\zref@extractdefault{#1}{anchor}{}}%
2308       \zref@ifrefcontainsprop{#1}{urluse}{%
2309         {\zref@extractdefault{#1}{urluse}{}}%
2310       }{%
2311         {\zref@extractdefault{#1}{url}{}}%
2312       }%
2313     }{}%
2314   }%

```

```

2315 }%
2316 }

```

`\ZREF@xr@zref@ignorewarning`

```

2317 \def\ZREF@xr@zref@ignorewarning#1{%
2318   \PackageWarningNoLine{zref-xr}{%
2319     Zref label ‘#1’ is already in use\MessageBreak
2320     in file ‘\ZREF@xr@file’%
2321   }%
2322   \edef\ZREF@xr@ignored@zref{%
2323     \the\numexpr\ZREF@xr@ignored@zref+1%
2324   }%
2325 }%

```

`\ZREF@xr@ltx@ignorewarning`

```

2326 \def\ZREF@xr@ltx@ignorewarning#1{%
2327   \PackageWarningNoLine{zref-xr}{%
2328     LaTeX label ‘#1’ is already in use\MessageBreak
2329     in file ‘\ZREF@xr@file’%
2330   }%
2331   \edef\ZREF@xr@ignored@ltx{%
2332     \the\numexpr\ZREF@xr@ignored@ltx+1%
2333   }%
2334 }%

```

`\ZREF@xr@checklist`

```

2335 \def\ZREF@xr@checklist#1#2#3\ZREF@nil{%
2336   \ifx\@undefined#1\relax
2337     \expandafter\ZREF@xr@checkkey\string#1\@nil
2338   \fi
2339   \ifx\\#3\\%
2340     \else
2341       \ltx@ReturnAfterFi{%
2342         \ZREF@xr@checklist#3\ZREF@nil
2343       }%
2344     \fi
2345   }%
2346   \def\ZREF@xr@checkkey#1#2\@nil{%
2347     \zref@ifpropundefined{#2}{%
2348       \zref@newprop{#2}{}%
2349     }{}%
2350   }%

```

`\ZREF@xr@scanparams`

```

2351 \def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{%
2352   \let#1\ltx@empty
2353   \ZREF@foundfalse
2354   \ZREF@xr@scantitleref#1#2\TR@TitleReference{}{}\ZREF@nil
2355   \ifZREF@found
2356     \else
2357       \ltx@LocalAppendToMacro#1{\default{#2}}%
2358     \fi
2359   % page
2360   \ltx@LocalAppendToMacro#1{\page{#3}}%
2361   % nameref title
2362   \ifZREF@found
2363     \else
2364       \ifx\\#4\\%

```

```

2365 \else
2366 \def\ZREF@xr@temp{#4}%
2367 \ifx\ZREF@xr@temp\ZREF@xr@relax
2368 \else
2369 \ltx@LocalAppendToMacro#1{\title{#4}}%
2370 \fi
2371 \fi
2372 \fi
2373 % anchor
2374 \ifx\#5\%
2375 \else
2376 \ltx@LocalAppendToMacro#1{\anchor{#5}}%
2377 \fi
2378 \ifx\#6\%
2379 \else
2380 \ifZREF@xr@urluse
2381 \ZREF@xr@AddURL{#6}%
2382 \expandafter\ltx@LocalAppendToMacro\expandafter#1%
2383 \expandafter{%
2384 \expandafter\urluse\expandafter{%
2385 \csname Z@U@\ZREF@xr@theURL\endcsname
2386 }%
2387 }%
2388 \else
2389 \ltx@LocalAppendToMacro#1{\url{#6}}%
2390 \fi
2391 \fi
2392 }%

```

\ZREF@xr@scantitleref

```

2393 \def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{%
2394 \ifx\#5\%
2395 \else
2396 \ltx@LocalAppendToMacro#1{%
2397 \default{#3}%
2398 \title{#4}%
2399 }%
2400 \ZREF@foundtrue
2401 \fi
2402 }%

```

\ZREF@xr@urlcheck

```

2403 \def\ZREF@xr@urlcheck#1{%
2404 \zref@ifrefcontainsprop{#1}{anchor}{%
2405 \zref@ifrefcontainsprop{#1}{url}{%
2406 }{%
2407 \expandafter
2408 \ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
2409 \expandafter{%
2410 \csname url\ifZREF@xr@urluse use\fi
2411 \expandafter\endcsname\expandafter{\ZREF@xr@url}%
2412 }%
2413 }%
2414 }{%
2415 }%
2416 }%
2417 \</xr>

```

6.17 Module hyperref

```
UNFINISHED :-(
2418 (*hyperref)
2419 \NeedsTeXFormat{LaTeX2e}
2420 \ProvidesPackage{zref-hyperref}%
2421   [2016/05/16 v2.25 Module hyperref for zref (H0)]%
2422 \RequirePackage{zref-base}[2016/05/16]
2423 \ifx\ZREF@base@ok Y%
2424 \else
2425   \expandafter\endinput
2426 \fi

2427 \ZREF@NewPropAnchor
2428 \zref@addprop\ZREF@mainlist{anchor}%
2429 /hyperref)
```

6.18 Module savepos

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

6.18.1 Identification

```
2430 (*savepos)
2431 \NeedsTeXFormat{LaTeX2e}
2432 \ProvidesPackage{zref-savepos}%
2433   [2016/05/16 v2.25 Module savepos for zref (H0)]%
2434 \RequirePackage{zref-base}[2016/05/16]
2435 \ifx\ZREF@base@ok Y%
2436 \else
2437   \expandafter\endinput
2438 \fi
```

6.18.2 Availability

First we check, whether the feature is available.

```
2439 \ltx@ifundefined{pdfsavepos}{%
2440   \PackageError\ZREF@name{%
2441     \string\pdfsavepos\space is not supported.\MessageBreak
2442     It is provided by pdfTeX (1.40) or XeTeX%
2443   }\ZREF@UpdatePdfTeX
2444   \endinput
2445 }{}
```

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`.

```
2446 \RequirePackage{ifpdf}
2447 \ifpdf
2448 \else
2449   \ltx@ifundefined{pdftexversion}{%
2450     }{%
2451       \ifnum\pdftexversion<140 %
2452         \PackageError\ZREF@name{%
2453           \string\pdfsavepos\space is not supported in DVI mode%
2454           \MessageBreak
2455           of this pdfTeX version%
2456         }\ZREF@UpdatePdfTeX
```

```

2457     \expandafter\expandafter\expandafter\endinput
2458     \fi
2459 }%
2460 \fi

```

6.18.3 Setup

```

2461 \zref@newlist{savepos}
2462 \zref@newprop*{posx}[0]{\the\pdflastxpos}
2463 \zref@newprop*{posy}[0]{\the\pdflastypos}
2464 \zref@addprops{savepos}{posx,posy}

```

6.18.4 User macros

`\zref@savepos`

```

2465 \def\zref@savepos{%
2466   \if@files
2467     \pdfsavepos
2468   \fi
2469 }

```

`\ZREF@zsavapos`

```

2470 \def\ZREF@zsavapos#1#2#3{%
2471   \@bsphack
2472   \if@files
2473     \zref@savepos
2474     #1{#3}{#2}%
2475     \ltx@ifundefined{TeXeTstate}{%
2476       {%
2477         \ifnum\TeXeTstate=\ltx@zero
2478         \else
2479           \zref@savepos
2480         \fi
2481       }%
2482     \fi
2483   \@esphack
2484 }

```

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

```

2485 \ZREF@ifDefinable\zsavapos\def{%
2486   {%
2487     \ZREF@zsavapos\zref@labelbylist{savepos}%
2488   }%
2489 }

```

`\zsavaposx`

```

2490 \ZREF@ifDefinable\zsavaposx\def{%
2491   {%
2492     \ZREF@zsavapos\zref@labelbyprops{posx}%
2493   }%
2494 }

```

`\zsavaposy`

```

2495 \ZREF@ifDefinable\zsavaposy\def{%
2496   {%
2497     \ZREF@zsavapos\zref@labelbyprops{posy}%
2498   }%
2499 }

```

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

```
2500 \newcommand*{\zposx}[1]{%
2501   \zref@extract{#1}{posx}%
2502 }%
2503 \newcommand*{\zposy}[1]{%
2504   \zref@extract{#1}{posy}%
2505 }%
```

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

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

`\ZREF@savepos@ok`

```
2506 \let\ZREF@savepos@ok=Y
2507 </savepos>
```

6.19 Module `abspos`

6.19.1 Identification

```
2508 (*abspos)
2509 \NeedsTeXFormat{LaTeX2e}
2510 \ProvidesPackage{zref-abspos}%
2511   [2016/05/16 v2.25 Module abspos for zref (H0)]%
2512 \RequirePackage{zref-base}[2016/05/16]
2513 \ifx\ZREF@base@ok Y%
2514 \else
2515   \expandafter\endinput
2516 \fi
2517 \RequirePackage{zref-savepos}[2016/05/16]
2518 \ifx\ZREF@savepos@ok Y%
2519 \else
2520   \expandafter\endinput
2521 \fi
2522 \RequirePackage{zref-pagelayout}[2016/05/16]
2523 \zref@addprop{savepos}{abspage}
2524 \RequirePackage{ifpdf}[2010/09/13]
```

`\zref@absposx`

```
2525 \newcommand*{\zref@absposx}[3]{%
2526   \number
2527   \expandafter\zref@absposnumx\expandafter{%
2528     \number\zref@extractdefault{#1}{abspage}{0}%
2529   }{#2}{#3}%
2530   \ltx@space
2531 }
```

`\zref@absposy`

```
2532 \newcommand*{\zref@absposy}[3]{%
2533   \number
2534   \expandafter\zref@absposnumy\expandafter{%
2535     \number\zref@extractdefault{#1}{abspage}{0}%
2536   }{#2}{#3}%
2537 }
```

```

2536 }{#2}{#3}%
2537 \ltx@space
2538 }

\zref@absposnumx
2539 \newcommand*{\zref@absposnumx}[3]{%
2540   \number
2541   % \ifnum#1>\ltx@zero
2542   % \zref@ifrefundefined{thepage#1}{%
2543   %   0%
2544   % }{%
2545   % \numexpr\ZREF@absposnum{thepage#1}{#2}{x}{#3}\relax
2546   % }%
2547   % \else
2548   %   0%
2549   % \fi
2550 }

\zref@absposnumy
2551 \newcommand*{\zref@absposnumy}[3]{%
2552   \number
2553   % \ifnum#1>\ltx@zero
2554   % \zref@ifrefundefined{thepage#1}{%
2555   %   0%
2556   % }{%
2557   % \numexpr\ZREF@absposnum{thepage#1}{#2}{y}{#3}\relax
2558   % }%
2559   % \else
2560   %   0%
2561   % \fi
2562 }

\ZREF@absposnum
2563 \def\ZREF@absposnum#1#2#3#4{%
2564   \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
2565     0%
2566   }{%
2567     \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
2568   }%
2569 }

\zref@def@absposx
2570 \ZREF@Robust\def\zref@def@absposx#1{%
2571   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposx}%
2572 }

\zref@def@absposy
2573 \ZREF@Robust\def\zref@def@absposy#1{%
2574   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
2575 }

\zref@def@absposnumx
2576 \ZREF@Robust\def\zref@def@absposnumx#1{%
2577   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumx}%
2578 }

```

```

\zref@def@absposnumy
2579 \ZREF@Robust\def\zref@def@absposnumy#1{%
2580   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumy}%
2581 }

\ZREF@def@abspos
2582 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2583   \edef#1{#2{#3}{#4}{#5}}%
2584 }

\zref@absposused
2585 \ZREF@Robust\def\zref@absposused{%
2586   \zref@wrapper@babel\ZREF@abspos@used
2587 }

\ZREF@abspos@used
2588 \def\ZREF@abspos@used#1{%
2589   \zref@refused{#1}%
2590   \zref@ifrefundefined{#1}{%
2591     }{%
2592     \begingroup
2593       \edef\ZREF@temp{%
2594         \zref@extractdefault{#1}{abspage}{0}%
2595       }%
2596       \ifnum\ZREF@temp>\ltx@zero
2597         \zref@refused{thepage\ZREF@temp}%
2598       \else
2599         \@PackageError{zref-abspos}{%
2600           \string\zref@pos@label@used\ltx@space
2601           needs property 'abspage'\MessageBreak
2602           in label '#1'%
2603         }\@ehc
2604       \fi
2605     \endgroup
2606   }%
2607 }

\zref@absposnumused
2608 \newcommand*{\zref@absposnumused}[1]{%
2609   \ifnum#1>\ltx@zero
2610     \zref@refused{thepage\number#1}%
2611   \else
2612     \@PackageError{zref-abspos}{%
2613       Invalid absolute page number (#1)\MessageBreak
2614       for \string\zref@pos@num@used.\MessageBreak
2615       A positive integer is expected%
2616     }\@ehc
2617   \fi
2618 }

\zref@ifabsposundefined
2619 \def\zref@ifabsposundefined#1{%
2620   \zref@ifrefundefined{#1}\ltx@firsttwo{%
2621     \expandafter\zref@ifabsposnumundefined\expandafter{%
2622       \number\zref@extractdefault{#1}{abspage}{0}%
2623     }%
2624   }%
2625 }

```

\zref@ifabsposnumundefined

```
2626 \def\zref@ifabsposnumundefined#1{%
2627   \ifnum\ZREF@number{#1}>\lt\tx@zero
2628     \zref@ifrefundefined{thepage#1}%
2629     \lt\tx@firstoftwo\lt\tx@secondoftwo
2630   \else
2631     \expandafter\lt\tx@firstoftwo
2632   \fi
2633 }
```

6.19.2 Media

\ZREF@abspos@media@width

```
2634 \edef\ZREF@abspos@media@width{%
2635   \lt\tx@ifundefined{pdfpagewidth}{%
2636     \lt\tx@ifundefined{mediawidth}{%
2637       \lt\tx@ifundefined{stockwidth}{%
2638         paperwidth%
2639       }{%
2640         stockwidth%
2641       }%
2642     }{%
2643       mediawidth%
2644     }%
2645   }{%
2646     pdfpagewidth%
2647   }%
2648 }
```

\ZREF@abspos@media@height

```
2649 \edef\ZREF@abspos@media@height{%
2650   \lt\tx@ifundefined{pdfpageheight}{%
2651     \lt\tx@ifundefined{mediaheight}{%
2652       \lt\tx@ifundefined{stockheight}{%
2653         paperheight%
2654       }{%
2655         stockheight%
2656       }%
2657     }{%
2658       mediaheight%
2659     }%
2660   }{%
2661     \noexpand\ifcase\pdfpageheight
2662       \lt\tx@ifundefined{stockheight}{%
2663         paperheight%
2664       }{%
2665         stockheight%
2666       }%
2667     \noexpand\else
2668       pdfpageheight%
2669     \noexpand\fi
2670   }%
2671 }
```

\ZREF@abspos@media@x@left

```
2672 \def\ZREF@abspos@media@x@left#1{%
2673   0%
2674 }
```

```

\ZREF@abspos@media@x@right
2675 \def\ZREF@abspos@media@x@right#1{%
2676   \zref@extract{#1}\ZREF@abspos@media@width
2677 }

\ZREF@abspos@media@x@center
2678 \def\ZREF@abspos@media@x@center#1{%
2679   \ZREF@abspos@media@x@left{#1}%
2680   +\zref@extract{#1}\ZREF@abspos@media@width/2%
2681 }

\ZREF@abspos@media@y@top
2682 \def\ZREF@abspos@media@y@top#1{%
2683   \zref@extract{#1}\ZREF@abspos@media@height
2684 }

\ZREF@abspos@media@y@bottom
2685 \def\ZREF@abspos@media@y@bottom#1{%
2686   0%
2687 }

\ZREF@abspos@media@y@center
2688 \def\ZREF@abspos@media@y@center#1{%
2689   \zref@extract{#1}\ZREF@abspos@media@height/2%
2690 }

```

6.19.3 Paper

```

\ZREF@abspos@paper@x@left
2691 \def\ZREF@abspos@paper@x@left#1{%
2692   0%
2693 }

\ZREF@abspos@paper@x@right
2694 \def\ZREF@abspos@paper@x@right#1{%
2695   \zref@extract{#1}{paperwidth}%
2696 }

\ZREF@abspos@paper@x@center
2697 \def\ZREF@abspos@paper@x@center#1{%
2698   \zref@extract{#1}{paperwidth}/2%
2699 }

\ZREF@abspos@paper@y@top
2700 \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top

\ZREF@abspos@paper@y@bottom
2701 \def\ZREF@abspos@paper@y@bottom#1{%
2702   \ZREF@abspos@paper@y@top{#1}%
2703   -\zref@extract{#1}{paperheight}%
2704 }

\ZREF@abspos@paper@y@center
2705 \def\ZREF@abspos@paper@y@center#1{%
2706   \ZREF@abspos@paper@y@top{#1}%
2707   -\zref@extract{#1}{paperheight}/2%
2708 }

```

6.19.4 Origin

`\ZREF@abspos@origin@x`

```
2709 \let\ZREF@temp\ltx@two
2710 \ltx@ifundefined{pdfhorigin}{}{%
2711   \ifpdf
2712     \let\ZREF@temp\ltx@zero
2713   \fi
2714 }
2715 \ifx\ZREF@temp\ltx@two
2716   \ifnum\mag=1000 %
2717     \let\ZREF@temp\ltx@one
2718   \fi
2719 \fi
2720 \ifcase\ZREF@temp
2721   \def\ZREF@abspos@origin@x#1{%
2722     \zref@extract{#1}{pdfhorigin}%
2723   }%
2724 \or
2725   \def\ZREF@abspos@origin@x#1{%
2726     4736286%
2727   }%
2728 \or
2729   \def\ZREF@abspos@origin@x#1{%
2730     \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2731   }%
2732 \fi
```

`\ZREF@abspos@origin@y`

```
2733 \let\ZREF@temp\ltx@two
2734 \ltx@ifundefined{pdfvorigin}{}{%
2735   \ifpdf
2736     \let\ZREF@temp\ltx@zero
2737   \fi
2738 }
2739 \ifx\ZREF@temp\ltx@two
2740   \ifnum\mag=1000 %
2741     \let\ZREF@temp\ltx@one
2742   \fi
2743 \fi
2744 \ifcase\ZREF@temp
2745   \def\ZREF@abspos@origin@y#1{%
2746     \zref@extract{#1}{pdfvorigin}%
2747   }%
2748 \or
2749   \def\ZREF@abspos@origin@y#1{%
2750     4736286%
2751   }%
2752 \or
2753   \def\ZREF@abspos@origin@y#1{%
2754     \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2755   }%
2756 \fi
```

6.19.5 Header

`\ZREF@abspos@head@x@left`

```
2757 \def\ZREF@abspos@head@x@left#1{%
```

```

2758 \ZREF@abspos@paper@x@left{#1}%
2759 +\ZREF@abspos@origin@x{#1}%
2760 +\zref@extract{#1}{hoffset}%
2761 +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2762 \zref@extract{#1}{oddsidemargin}%
2763 \else
2764 \zref@extract{#1}{evensidemargin}%
2765 \fi
2766 }

```

\ZREF@abspos@head@x@right

```

2767 \def\ZREF@abspos@head@x@right#1{%
2768 \ZREF@abspos@head@x@left{#1}%
2769 +\zref@extract{#1}{textwidth}%
2770 }

```

\ZREF@abspos@head@x@center

```

2771 \def\ZREF@abspos@head@x@center#1{%
2772 \ZREF@abspos@head@x@left{#1}%
2773 +\zref@extract{#1}{textwidth}/2%
2774 }

```

\ZREF@abspos@head@y@top

```

2775 \def\ZREF@abspos@head@y@top#1{%
2776 \ZREF@abspos@paper@y@top{#1}%
2777 -\ZREF@abspos@origin@y{#1}%
2778 -\zref@extract{#1}{voffset}%
2779 -\zref@extract{#1}{topmargin}%
2780 }

```

\ZREF@abspos@head@y@bottom

```

2781 \def\ZREF@abspos@head@y@bottom#1{%
2782 \ZREF@abspos@head@y@top{#1}%
2783 -\zref@extract{#1}{headheight}%
2784 }

```

\ZREF@abspos@head@y@center

```

2785 \def\ZREF@abspos@head@y@center#1{%
2786 \ZREF@abspos@head@y@top{#1}%
2787 -\zref@extract{#1}{headheight}/2%
2788 }

```

6.19.6 Body

\ZREF@abspos@body@x@left

```

2789 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left

```

\ZREF@abspos@body@x@right

```

2790 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right

```

\ZREF@abspos@body@x@center

```

2791 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center

```

\ZREF@abspos@body@y@top

```

2792 \def\ZREF@abspos@body@y@top#1{%
2793 \ZREF@abspos@head@y@bottom{#1}%
2794 -\zref@extract{#1}{headsep}%
2795 }

```

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

```
2796 \def\ZREF@abspos@body@y@bottom#1{%
2797   \ZREF@abspos@body@y@top{#1}%
2798   -\zref@extract{#1}{textheight}%
2799 }
```

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

```
2800 \def\ZREF@abspos@body@y@center#1{%
2801   \ZREF@abspos@body@y@top{#1}%
2802   -\zref@extract{#1}{textheight}/2%
2803 }
```

6.19.7 Footer

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

```
2804 \let\ZREF@abspos@foot@x@left\ZREF@abspos@head@x@left
```

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

```
2805 \let\ZREF@abspos@foot@x@right\ZREF@abspos@head@x@right
```

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

```
2806 \let\ZREF@abspos@foot@x@center\ZREF@abspos@head@x@center
```

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

```
2807 \def\ZREF@abspos@foot@y@bottom#1{%
2808   \ZREF@abspos@body@y@bottom{#1}%
2809   -\zref@extract{#1}{footskip}%
2810 }
```

6.19.8 Marginal notes

`\ZREF@abspos@marginpar@x@left`

```
2811 \def\ZREF@abspos@marginpar@x@left#1{%
2812   \ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2813   \ZREF@abspos@body@x@right{#1}%
2814   +\zref@extract{#1}{marginparsep}%
2815   \else
2816   \ZREF@abspos@body@x@left{#1}%
2817   -\zref@extract{#1}{marginparsep}%
2818   -\zref@extract{#1}{marginparwidth}%
2819   \fi
2820 }
```

`\ZREF@abspos@marginpar@x@right`

```
2821 \def\ZREF@abspos@marginpar@x@right#1{%
2822   \ZREF@abspos@marginpar@x@left{#1}%
2823   +\zref@extract{#1}{marginparwidth}%
2824 }
```

`\ZREF@abspos@marginpar@x@center`

```
2825 \def\ZREF@abspos@marginpar@x@center#1{%
2826   \ZREF@abspos@marginpar@x@left{#1}%
2827   +\zref@extract{#1}{marginparwidth}/2%
2828 }
```

`\ZREF@abspos@marginpar@y@top`

```
2829 \let\ZREF@abspos@marginpar@y@top\ZREF@abspos@body@y@top
```

```

\ZREF@abspos@marginpar@y@bottom
2830 \let\ZREF@abspos@marginpar@y@bottom\ZREF@abspos@body@y@bottom

\ZREF@abspos@marginpar@y@center
2831 \let\ZREF@abspos@marginpar@y@center\ZREF@abspos@body@y@center

6.19.9 Stock paper

\ZREF@abspos@stock@x@left
2832 \let\ZREF@abspos@stock@x@left\ZREF@abspos@paper@x@left

\ZREF@abspos@stock@x@right
2833 \let\ZREF@abspos@stock@x@right\ZREF@abspos@paper@x@right

\ZREF@abspos@stock@x@center
2834 \let\ZREF@abspos@stock@x@center\ZREF@abspos@paper@x@center

\ZREF@abspos@stock@y@top
2835 \let\ZREF@abspos@stock@y@top\ZREF@abspos@paper@y@top

\ZREF@abspos@stock@y@bottom
2836 \let\ZREF@abspos@stock@y@bottom\ZREF@abspos@paper@y@bottom

\ZREF@abspos@stock@y@center
2837 \let\ZREF@abspos@stock@y@center\ZREF@abspos@paper@y@center

2838 (/abspos)

```

6.20 Module **dotfill**

```

2839 (*dotfill)
2840 \NeedsTeXFormat{LaTeX2e}
2841 \ProvidesPackage{zref-dotfill}%
2842 [2016/05/16 v2.25 Module dotfill for zref (HO)]%
2843 \RequirePackage{zref-base}[2016/05/16]
2844 \ifx\ZREF@base@ok Y%
2845 \else
2846 \expandafter\endinput
2847 \fi

```

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

```
2848 \RequirePackage{zref-savepos}[2016/05/16]
```

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

```
2849 \zref@require@unique
```

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

```
2850 \RequirePackage{keyval}
```

The definitions of the keys follow.

```

2851 \define@key{ZREF@DF}{unit}{%
2852   \def\ZREF@df@unit{#1}%
2853 }
2854 \define@key{ZREF@DF}{min}{%
2855   \def\ZREF@df@min{#1}%
2856 }
2857 \define@key{ZREF@DF}{dot}{%

```

```

2858 \def\ZREF@df@dot{#1}%
2859 }
Defaults are set, see user interface.
2860 \providecommand\ZREF@df@min{2}
2861 \providecommand\ZREF@df@unit{.44em}
2862 \providecommand\ZREF@df@dot{.}
\zdotfillsetup Configuration of \zdotfill is done by \zdotfillsetup.
2863 \newcommand*{\zdotfillsetup}{\kvsetkeys{ZREF@DF}}

\zdotfill \zdotfill sets labels at the left and the right to get the horizontal position.
\zsavapos is not used, because we do not need the vertical position.
2864 \ZREF@IfDefinable\zdotfill\def{%
2865   {%
2866     \leavevmode
2867     \global\advance\c@zref@unique\ltx@one
2868     \begingroup
2869     \def\ZREF@temp{zref@\number\c@zref@unique}%
2870     \pdfsavapos
2871     \zref@labelbyprops{\thezref@unique L}{posx}%
2872     \setlength{\dimen@}{\ZREF@df@unit}%
2873     \zref@ifrefundefined{\thezref@unique R}{%
2874       \ZREF@dotfill
2875     }{%
2876       \ifnum\numexpr\zposx{\thezref@unique R}%
2877         -\zposx{\thezref@unique L}\relax
2878         <\dimexpr\ZREF@df@min\dimen@\relax
2879         \hfill
2880       \else
2881         \ZREF@dotfill
2882       \fi
2883     }%
2884     \pdfsavapos
2885     \zref@labelbyprops{\thezref@unique R}{posx}%
2886     \endgroup
2887     \kern\z@
2888   }%
2889 }

\ZREF@dotfill Help macro that actually sets the dots.
2890 \def\ZREF@dotfill{%
2891   \cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill
2892 }

2893 </dotfill>

```

6.21 Module env

```

2894 (*env)
2895 \NeedsTeXFormat{LaTeX2e}
2896 \ProvidesPackage{zref-env}%
2897 [2016/05/16 v2.25 Module env for zref (HO)]%
2898 \RequirePackage{zref-base}[2016/05/16]
2899 \ifx\ZREF@base@ok Y%
2900 \else
2901   \expandafter\endinput
2902 \fi
2903 \zref@newprop{envname}[]{\@currentenv}

```

```

2904 \zref@newprop{envline}[]{\zref@env@line}

\zref@env@line Macro \zref@env@line extracts the line number from \@currentvline.
2905 \def\zref@env@line{%
2906   \ifx\@currentvline\ltx@empty
2907   \else
2908     \expandafter
2909     \ZREF@ENV@line\@currentvline\ltx@empty line \ltx@empty\@nil
2910   \fi
2911 }

\ZREF@ENV@line
2912 \def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%
2913 </env>

```

7 Test

7.1 \zref@localaddprop

```

2914 <test1>
2915 \NeedsTeXFormat{LaTeX2e}
2916 \nofiles
2917 \documentclass{article}
2918 \usepackage{zref-base}[2016/05/16]
2919 \usepackage{qstest}
2920 \IncludeTests{*}
2921 \LogTests{log}{*}{*}
2922
2923 \makeatletter
2924 \def\ExpectList#1#2{%
2925   \expandafter\expandafter\expandafter\Expect
2926   \expandafter\expandafter\expandafter{\csname Z@L@#1\endcsname}{#2}%
2927 }
2928 \begin{qstest}{\localaddprop}{\localaddprop}
2929   \ExpectList{main}{\default\page}%
2930   \Expect{undefined}*\meaning\foobar}%
2931   \zref@newprop{foobar}{F00}%
2932   \Expect{undefined}*\meaning\foobar}%
2933   \zref@newlist{alist}%
2934   \ExpectList{alist}{}%
2935   \begingroup
2936     \zref@localaddprop{main}{foobar}%
2937     \Expect{undefined}*\meaning\foobar}%
2938     \ExpectList{main}{\default\page\foobar}%
2939     \zref@localaddprop{alist}{page}%
2940     \ExpectList{alist}{\page}%
2941   \endgroup
2942   \ExpectList{main}{\default\page}%
2943   \ExpectList{alist}{}%
2944   \zref@addprop{alist}{foobar}%
2945   \ExpectList{alist}{\foobar}%
2946   \Expect{undefined}*\meaning\foobar}%
2947 \end{qstest}
2948 \@@end
2949 </test1>

```

7.2 Module base

```

2950 (*test-base)
2951 \NeedsTeXFormat{LaTeX2e}
2952 \documentclass{article}
2953 \usepackage{zref-base,zref-titleref}[2016/05/16]
2954 \usepackage{qstest}
2955 \IncludeTests{*}
2956 \LogTests{log}{*}{*}
2957
2958 \makeatletter
2959 \newcommand*\DefExpand}[2]{%
2960   \expandafter\expandafter\expandafter\def
2961     \expandafter\expandafter\expandafter#1%
2962     \expandafter\expandafter\expandafter{#2}%
2963   \@onelevel@sanitize#1%
2964 }
2965 \newcommand*\Test}[3]{%
2966   \Expect{#2}*{#1}%
2967   \zref@wrapper@unexpanded{%
2968     \Expect*{#3}*{#1}%
2969   }%
2970   \DefExpand\x{#1}%
2971   \Expect*{#3}*{\x}%
2972 }
2973 \makeatother
2974
2975 \begin{document}
2976 \section{\textit{Hello} \textbf{World}}
2977 \label{sec:hello}
2978 \makeatletter
2979 \zref@newprop{foo}[\@empty D\@empty efault]{\@empty V\@empty alue}
2980 \begin{qstest}{getcurrent}{getcurrent}
2981   \Test{\zref@getcurrent{foo}}%
2982     {Value}{\noexpand\@empty V\noexpand\@empty alue}%
2983   \Test{\zref@getcurrent{xy}}{-}{-}%
2984 \end{qstest}
2985 \begin{qstest}{extract}{extract}
2986   \def\textbf#1{<#1>}%
2987   \def\textit#1{[#1]}% hash-ok
2988   \Test{\zref@extractdefault{xy}{page}{\@empty D\@empty efault}}%
2989     {Default}{\noexpand\@empty D\noexpand\@empty efault}%
2990   \Test{\zref@extractdefault{sec:hello}{foo}{\@empty A\@empty B}}%
2991     {AB}{\noexpand\@empty A\noexpand\@empty B}%
2992   \Test{\zref@extract{sec:hello}{foo}}%
2993     {Default}{\noexpand\@empty D\noexpand\@empty efault}%
2994   \zref@ifrefundefined{sec:hello}{%
2995   }{%
2996     \Test{\zref@extract{sec:hello}{default}}{1}{1}%
2997     \Test{\zref@extract{sec:hello}{title}}%
2998       {[Hello] <World>}%
2999     {\noexpand\textit{Hello} \noexpand\textbf{World}}%
3000   }%
3001 \end{qstest}
3002 \end{document}
3003 \end{test-base}

```

7.3 Module runs

```

3004 (*test-runs)
3005 \NeedsTeXFormat{LaTeX2e}

```

```

3006 \documentclass{article}
3007 \usepackage{zref-runs}[2016/05/16]
3008 \usepackage{qstest}
3009 \IncludeTests{*}
3010 \LogTests{log}{*}{*}
3011
3012 \begin{qstest}{zruns-preamble}{zruns-preamble}
3013   \Expect{0}{*}{\zruns}%
3014 \end{qstest}
3015
3016 \AtBeginDocument{%
3017   \begin{qstest}{zruns-atbegindocument}{zruns-atbegindocument}%
3018     \Expect*{\number\ExpectRuns}{*}{\zruns}%
3019   \end{qstest}%
3020 }
3021
3022 \begin{document}
3023 \begin{qstest}{zruns-document}{zruns-document}
3024   \Expect*{\number\ExpectRuns}{*}{\zruns}%
3025 \end{qstest}
3026 \end{document}
3027 /test-runs)

```

7.4 Module titleref

```

3028 (*test-titleref-memoir)
3029 \NeedsTeXFormat{LaTeX2e}
3030 \documentclass{memoir}
3031 \usepackage{zref-titleref}[2016/05/16]
3032 \usepackage{qstest}
3033 \IncludeTests{*}
3034 \LogTests{log}{*}{*}
3035 \begin{document}
3036 \makeatletter
3037 \def\List{}
3038 \def\Label#1{%
3039   \zref@label{#1}%
3040   \g@addto@macro\List{%
3041     \par
3042     #1: [\ztitleref{#1}]%
3043   }%
3044   \mbox{}%
3045   \zref@refused{#1}%
3046   \zref@ifrefundefined{#1}{%
3047   }{%
3048     \begingroup
3049       \edef\x{\zref@extract{#1}{title}}%
3050       \Expect{OK/}{*}{\expandafter\ltx@carthree\x{}{}{}\@nil}%
3051     \endgroup
3052   }%
3053 }
3054 \def\Test#1{%
3055   \csname#1\endcsname*{OK/#1}%
3056   \Label{#1*}%
3057   \csname#1\endcsname{OK/#1}%
3058   \Label{#1}%
3059   \csname#1\endcsname[OK/#1-toc]%
3060   {WRONG-in-titleref/#1-toc-2}%
3061   \Label{#1-toc}%

```

```

3062 \expandafter\ifx\csname#1\endcsname\part
3063 \else
3064   \headnamereffalse
3065   \csname#1\endcsname[OK/#1-th-toc]%
3066                               [WRONG-in-titleref/#1-th-toc-2]%
3067                               {WRONG-in-titleref/#1-th-toc-3}%
3068   \Label{#1-th-toc}%
3069   \headnamerefftrue
3070   \csname#1\endcsname[WRONG-in-titleref/#1-th-head-1]%
3071                               [OK/#1-th-head]%
3072                               {WRONG-in-titleref/#1-th-head-3}%
3073   \Label{#1-th-head}%
3074 \fi
3075 }
3076 \begin{qstest}{section}{section}
3077   \@for\x:=part,chapter,section,subsection,subsubsection\do{%
3078     \expandafter\Test\expandafter\x}%
3079   }%
3080 \end{qstest}
3081 \newpage
3082 \List
3083 \end{document}
3084 /test-titleref-memoir)

```

8 Installation

8.1 Download

Package. This package is available on CTAN²:

[CTAN:macros/latex/contrib/oberdiek/zref.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/zref.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ 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/oberdiek.tds.zip](#)

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

8.2 Bundle installation

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

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

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

²<http://ctan.org/pkg/zref>

8.3 Package installation

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

```
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/oberdiek/zref.sty</code>
<code>zref-base.sty</code>	<code>→ tex/latex/oberdiek/zref-base.sty</code>
<code>zref-abspage.sty</code>	<code>→ tex/latex/oberdiek/zref-abspage.sty</code>
<code>zref-abspos.sty</code>	<code>→ tex/latex/oberdiek/zref-abspos.sty</code>
<code>zref-counter.sty</code>	<code>→ tex/latex/oberdiek/zref-counter.sty</code>
<code>zref-dotfill.sty</code>	<code>→ tex/latex/oberdiek/zref-dotfill.sty</code>
<code>zref-env.sty</code>	<code>→ tex/latex/oberdiek/zref-env.sty</code>
<code>zref-hyperref.sty</code>	<code>→ tex/latex/oberdiek/zref-hyperref.sty</code>
<code>zref-lastpage.sty</code>	<code>→ tex/latex/oberdiek/zref-lastpage.sty</code>
<code>zref-marks.sty</code>	<code>→ tex/latex/oberdiek/zref-marks.sty</code>
<code>zref-nextpage.sty</code>	<code>→ tex/latex/oberdiek/zref-nextpage.sty</code>
<code>zref-pageattr.sty</code>	<code>→ tex/latex/oberdiek/zref-pageattr.sty</code>
<code>zref-pagelayout.sty</code>	<code>→ tex/latex/oberdiek/zref-pagelayout.sty</code>
<code>zref-perpage.sty</code>	<code>→ tex/latex/oberdiek/zref-perpage.sty</code>
<code>zref-runs.sty</code>	<code>→ tex/latex/oberdiek/zref-runs.sty</code>
<code>zref-savepos.sty</code>	<code>→ tex/latex/oberdiek/zref-savepos.sty</code>
<code>zref-thepage.sty</code>	<code>→ tex/latex/oberdiek/zref-thepage.sty</code>
<code>zref-titleref.sty</code>	<code>→ tex/latex/oberdiek/zref-titleref.sty</code>
<code>zref-totpages.sty</code>	<code>→ tex/latex/oberdiek/zref-totpages.sty</code>
<code>zref-user.sty</code>	<code>→ tex/latex/oberdiek/zref-user.sty</code>
<code>zref-xr.sty</code>	<code>→ tex/latex/oberdiek/zref-xr.sty</code>
<code>zref.pdf</code>	<code>→ doc/latex/oberdiek/zref.pdf</code>
<code>zref-example.tex</code>	<code>→ doc/latex/oberdiek/zref-example.tex</code>
<code>zref-example-lastpage.tex</code>	<code>→ doc/latex/oberdiek/zref-example-lastpage.tex</code>
<code>zref-example-nextpage.tex</code>	<code>→ doc/latex/oberdiek/zref-example-nextpage.tex</code>
<code>test/zref-test1.tex</code>	<code>→ doc/latex/oberdiek/test/zref-test1.tex</code>
<code>test/zref-test-base.tex</code>	<code>→ doc/latex/oberdiek/test/zref-test-base.tex</code>
<code>test/zref-test-runs.tex</code>	<code>→ doc/latex/oberdiek/test/zref-test-runs.tex</code>
<code>test/zref-test-titleref-memoir.tex</code>	<code>→ doc/latex/oberdiek/test/zref-test-titleref-memoir.tex</code>
<code>zref.dtx</code>	<code>→ source/latex/oberdiek/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.

8.4 Refresh file name databases

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

8.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk zref.pdf unpack_files output .
```

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 pdfL^AT_EX:

```
pdflatex zref.dtx
makeindex -s gind.ist zref.idx
pdflatex zref.dtx
makeindex -s gind.ist zref.idx
pdflatex zref.dtx
```

9 Catalogue

The following XML file can be used as source for the [T_EX Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `zref.xml`.

```
3085 (*catalogue)
3086 <?xml version='1.0' encoding='us-ascii'?>
3087 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
3088 <entry datestamp='$Date$' modifier='$Author$' id='zref'>
3089   <name>zref</name>
3090   <caption>A new reference scheme for LaTeX.</caption>
3091   <authorref id='auth:oberdiek' />
3092   <copyright owner='Heiko Oberdiek' year='2006-2012' />
3093   <license type='lppl1.3' />
3094   <version number='2.25' />
3095   <description>
3096     This package offers a means to remove the limitation, of only two
3097     properties, that is inherent in the way LaTeX's reference system
3098     works. The package implements an extensible referencing system,
3099     where properties are handled in a more flexible way. It provides
3100     an interface for macro programmers to access the new reference
3101     scheme and some applications that use it.
3102     <p/>
3103     The package is part of the <xref refid='oberdiek'>oberdiek</xref>
3104     bundle.
3105   </description>
3106   <documentation details='Package documentation'
3107     href='ctan:/macros/latex/contrib/oberdiek/zref.pdf' />
3108   <ctan file='true' path='/macros/latex/contrib/oberdiek/zref.dtx' />
3109   <miktex location='oberdiek' />
3110   <texlive location='oberdiek' />
3111   <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip' />
3112 </entry>
3113 </catalogue>
```

10 References

- [1] Package `footmisc`, Robin Fairbairns, 2004/01/23 v5.3a.[CTAN:macros/latex/contrib/footmisc/footmisc.dtx](#)
- [2] Package `hyperref`, Sebastian Rahtz, Heiko Oberdiek, 2006/08/16 v6.75c.[CTAN:macros/latex/contrib/hyperref/](#)

- [3] Package `lastpage`, Jeff Goldberg, 1994/06/25 v0.1b.[CTAN:macros/latex/contrib/lastpage/](#)
- [4] Package `nameref`, Sebastian Rahtz, Heiko Oberdiek, 2006/02/12 v2.24.[CTAN:macros/latex/contrib/hyperref/nameref.dtx](#)
- [5] Package `perpage`, David Kastrup, 2002/12/20 v1.0.[CTAN:macros/latex/contrib/bigfoot/perpage.dtx](#)
- [6] Package `titleref`, Donald Arsenau, 2001/04/05 v3.1.[CTAN:macros/latex/contrib/misc/titleref.sty](#)
- [7] Package `totpages`, Wilhelm Müller, 1999/07/14 v1.00.[CTAN:macros/latex/contrib/totpages/](#)
- [8] Package `xr`, David Carlisle, 1994/05/28 v5.02.[CTAN:macros/latex/required/tools/xr.pdf](#)
- [9] Package `xr-hyper`, David Carlisle, 2000/03/22 v6.00beta4.[CTAN:macros/latex/contrib/hyperref/xr-hyper.sty](#)

11 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 X_YTeX.

[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]

- `\zexternaldocument` fixed for property 'url' when importing `\new@label` (bug found by Victor Ivrii).
- Two expansion steps also in `\zref@wrapper@unexpanded`.
- Nested calls of `\zref@wrapper@unexpanded` possible.

[2010/04/28 v2.16]

- More consequent use of package 'ltxcmds' and 'hologo'.
- Module `pagelayout` added.
- Module `pageattr` added.
- Robustness introduced for non-expandable interface macros.
- Internal change of the data format of property lists (suggestion of Florent Chervet).
- Module `titleref`: Support of environment `description`.

[2010/05/01 v2.17]

- `\zref@newprop` throws an error if the property already exists.
- Module `xr`: Bug fix for the case of several `.aux` files (bug found by Victor Ivrii).
- Module `xr`: Property 'urluse' and option `urluse` added.

[2010/05/13 v2.18]

- Module `env` added.
- Module `savepos`: `\zref@savepos` added.

[2010/10/22 v2.19]

- `\zref@addprop` and `\zref@localaddprop` are limited to one property only (incompatibility to versions v2.13 to v2.18).
- `\zref@addprops` and `\zref@localaddprops` added.
- `\zref@delprop` and `\zref@localdelprop` added.
- `\zref@labelbykv` and `\zkvlabel` (module `user`) with keys `prop`, `list`, `delprop`, `immediate`, `values` added.

[2011/02/12 v2.20]

- Fix for warning in `zref-xr`.

[2011/03/18 v2.21]

- Fix in module `pagelayout` for `\zlistpagelayout`.
- Fix for `\zref@localaddprop` (probably since v2.19).

[2011/10/05 v2.22]

- Documentation fixed for `\zref@(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.

12 Index

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

Symbols	
<code>\@@end</code>	2948
<code>\@PackageError</code> ..	508, 524, 2599, 2612
<code>\@PackageInfo</code>	2062
<code>\@PackageInfoNoLine</code>	548, 563, 1258, 1346, 1440, 1453, 1465, 1521, 1566
<code>\@PackageWarning</code>	691
<code>\@addtoreset</code>	911, 1008
<code>\@auxout</code>	715
<code>\@begintheorem</code>	2001, 2006
<code>\@bsphack</code>	600, 610, 630, 2471
<code>\@caption</code>	1846
<code>\@chapter</code>	1858, 1895
<code>\@currentHref</code>	940
<code>\@currentlabel</code>	935
<code>\@currenvir</code>	2903
<code>\@currenvline</code>	2906, 2909
<code>\@ehc</code>	296, 306, 491, 514, 526, 1596, 2603, 2616
<code>\@empty</code>	2979, 2982, 2988, 2989, 2990, 2991, 2993
<code>\@esphack</code>	607, 627, 642, 2483
<code>\@for</code>	3077
<code>\@ifclassloaded</code>	1893, 1930
<code>\@ifdefinable</code>	242, 289
<code>\@ifnextchar</code>	530, 1712
<code>\@ifpackageloaded</code>	1950, 1969, 1977, 1996
<code>\@ifstar</code>	495, 2082
<code>\@ifundefined</code>	192, 909, 1723, 2218, 2260

<code>\@input</code>	2279	<code>\c@zpage</code>	1741, 1747
<code>\@inputcheck</code>		<code>\c@zref@unique</code> .	915, 1736, 2867, 2869
.....	2110, 2111, 2126, 2163, 2165	<code>\ch@pt@c</code>	1899, 1905, 1911
<code>\@latex@warning</code>	777	<code>\chapter</code>	24, 30, 32, 61, 82
<code>\@mainaux</code>	1681	<code>\ChapterPages</code>	91, 112
<code>\@namedef</code>	535	<code>\ChapterStart</code>	78, 135, 150, 166
<code>\@newl@bel</code>	285	<code>\ChapterStop</code>	85, 148, 165, 184
<code>\@nil</code>	1611, 1613,	<code>\chardef</code>	1159, 1174, 1183, 1187
.....	1811, 1818, 2000, 2001, 2154,	<code>\cleaders</code>	2891
.....	2157, 2337, 2346, 2909, 2912, 3050	<code>\cleardoublepage</code>	79, 86
<code>\@onelevel@sanitize</code>		<code>\clearpage</code>	62
.....	422, 440, 505, 533, 1999, 2002, 2963	<code>\closein</code>	2126
<code>\@opargbegintheorem</code>	1991	<code>\columnsep</code>	1309
<code>\@part</code>	1852	<code>\columnwidth</code>	1308
<code>\@schapter</code>	1876	<code>\comma@entry</code> ...	343, 344, 346, 352,
<code>\@sect</code>	1864	378, 379, 381, 387, 614, 616,
<code>\@spart</code>	1870	620, 1617, 1618, 1619, 1625, 1628
<code>\@ssect</code>	1882	<code>\comma@parse</code>	342, 377, 613, 1616
<code>\@testdef</code>	1328, 1329, 1500, 1501	<code>\count@</code>	1343, 1354, 1355,
<code>\@testopt</code>	2084, 2087, 2097	1357, 1392, 1393, 1402, 1404,
<code>\@tfor</code>	322, 720	1405, 1518, 1539, 1541, 1542,
<code>\@undefined</code>	1767, 2336	1547, 1548, 2052, 2057, 2068, 2069
<code>\@</code>	25, 26, 27,	<code>\csname</code>	252, 253, 290, 315, 316, 317,
.....	28, 153, 155, 157, 158, 170, 173,	326, 351, 352, 369, 370, 386,
.....	2245, 2339, 2364, 2374, 2378, 2394	387, 404, 405, 425, 427, 444,
<code>_</code>	44, 45	462, 465, 478, 536, 538, 539,
A		544, 554, 559, 565, 578, 587,
<code>\AddLineBeginAux</code>	280	604, 620, 651, 661, 730, 738,
<code>\advance</code>	744, 787, 788, 790, 810, 833,
.....	1053, 1393, 1548, 1736, 2058, 2867	834, 835, 896, 1333, 1505, 1646,
<code>\afterassignment</code>	233, 1137, 1141	1654, 1706, 1724, 1726, 1729,
<code>\AfterLastShipout</code> ..	1050, 1410, 1573	1731, 1743, 1749, 1754, 1755,
<code>\Alph</code>	7	1757, 1759, 1760, 1767, 1820,
<code>\anchor</code>	2376	1823, 2061, 2068, 2080, 2104,
<code>\AtBeginDocument</code>	2208, 2210, 2225, 2226, 2243,
.....	1029, 1226, 1675, 1844, 3016	2250, 2253, 2267, 2268, 2288,
<code>\AtBeginShipout</code>	1012, 1097	2385, 2408, 2410, 2567, 2926,
<code>\AtBeginShipoutAddToBox</code>	1098	3055, 3057, 3059, 3062, 3065, 3070
<code>\AtBeginShipoutBoxDepth</code>	1318	<code>\current@chapid</code>	80, 88
<code>\AtBeginShipoutBoxHeight</code>	1317	D	
<code>\AtBeginShipoutBoxWidth</code>	1316	<code>\DeclareBoolOption</code>	
<code>\AtEndOfPackage</code>	195	2037, 2038, 2039, 2043
<code>\AtVeryEndDocument</code>	1337, 1509	<code>\DeclareOption</code>	194
B		<code>\default</code> .	2357, 2397, 2929, 2938, 2942
<code>\beamer@section</code>	1932	<code>\DefExpand</code>	2959, 2970
<code>\beamer@subsection</code>	1938	<code>\define@key</code>	1819, 1822,
<code>\beamer@subsubsection</code>	1944	1825, 1828, 2040, 2851, 2854, 2857
<code>\begin</code>	23, 57, 100, 106, 156,	<code>\descriptionlabel</code>	1888
.....	172, 2928, 2975, 2980, 2985,	<code>\detokenize</code>	1806
.....	3012, 3017, 3022, 3023, 3035, 3076	<code>\dftest</code>	167, 174, 175,
<code>\bfseries</code>	928	176, 177, 178, 179, 180, 181, 182
C		<code>\dimen@</code>	2872, 2878, 2891
<code>\c@abspage</code>	1015, 1744, 1751	<code>\dimexpr</code>	153, 155, 1405, 2730, 2754, 2878
<code>\c@page</code> ...	954, 1053, 1741, 2761, 2812	<code>\directlua</code>	1250, 1432
		<code>\do</code>	327, 720, 3077
		<code>\documentclass</code>	4,
		39, 68, 272, 2917, 2952, 3006, 3030

<code>\dotfill</code>	169, 173	<code>\GetTitleStringExpand</code>	1801
		<code>\GetTitleStringNonExpand</code>	1803
		<code>\GetTitleStringResult</code>	1806
E			
<code>\emph</code>	150	H	
<code>\end</code>	34, 64, 130, 159, 183, 185, 2947, 2984, 3001, 3002, 3014, 3019, 3025, 3026, 3080, 3083	<code>\hb@xt@</code>	2891
<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, 1333, 1505, 1646, 1654, 1705, 1706, 1724, 1726, 1729, 1730, 1731, 1743, 1749, 1754, 1755, 1757, 1759, 1760, 1767, 1820, 1823, 2061, 2068, 2080, 2104, 2208, 2210, 2225, 2226, 2243, 2250, 2253, 2267, 2268, 2288, 2385, 2408, 2411, 2567, 2926, 3055, 3057, 3059, 3062, 3065, 3070	<code>\headheight</code>	1303
<code>\endinput</code>	192, 264, 277, 966, 1004, 1025, 1047, 1088, 1128, 1219, 1237, 1421, 1477, 1584, 1695, 1778, 2022, 2425, 2437, 2444, 2457, 2515, 2520, 2846, 2901	<code>\headmargin</code>	1315
<code>\escapechar</code>	329, 420, 460, 461, 467, 722, 1246, 1273, 1428	<code>\headnamereffalse</code>	3064
<code>\etex@unexpanded</code> ..	591, 819, 839, 2186	<code>\headnamereftrue</code>	3069
<code>\evensidemargin</code>	1300	<code>\headsep</code>	1304
<code>\Expect</code>	2925, 2930, 2932, 2937, 2946, 2966, 2968, 2971, 3013, 3018, 3024, 3050	<code>\hfill</code>	2879, 2891
<code>\ExpectList</code>	2924, 2929, 2934, 2938, 2940, 2942, 2943, 2945	<code>\hoffset</code>	1296
<code>\ExpectRuns</code>	3018, 3024	<code>\hss</code>	2891
<code>\externaldocument</code>	2212, 2254	I	
F			
<code>\fancyhead</code>	51, 54	<code>\if@filesw</code> 710, 1051, 1680, 2466, 2472	
<code>\fancyhf</code>	50, 53	<code>\if@safe@actives</code>	889
<code>\fancypagestyle</code>	52	<code>\ifcase</code> 115, 1195, 1617, 2661, 2720, 2744	
<code>\filename@area</code>	2187	<code>\ifcsname</code>	883, 1705, 1730
<code>\filename@parse</code>	2096	<code>\ifeof</code>	2111, 2165
<code>\foo</code>	18, 29, 31, 33	<code>\ifetex@unexpanded</code>	267
<code>\foobar</code> 2930, 2932, 2937, 2938, 2945, 2946		<code>\ifheadnameref</code>	1907, 1920
<code>\footskip</code>	1305	<code>\ifin@</code>	317
<code>\foremargin</code>	1312	<code>\ifluatex</code>	1242, 1424
<code>\frontmatter</code>	58, 103	<code>\ifnum</code>	476, 1065, 1170, 1180, 1186, 1243, 1392, 1425, 1547, 1589, 1642, 2057, 2134, 2138, 2143, 2451, 2477, 2541, 2553, 2596, 2609, 2627, 2716, 2740, 2876
G			
<code>\g@addto@macro</code> ..	350, 368, 1729, 3040	<code>\ifodd</code>	124, 2761, 2812
<code>\G@refundefinedtrue</code>	776	<code>\ifpdf</code>	2447, 2711, 2735
<code>\gdef</code>	412, 539, 544, 945, 1333, 1682, 1724, 1726	<code>\ifx</code>	437, 441, 474, 507, 565, 673, 676, 690, 729, 795, 964, 969, 976, 1002, 1023, 1045, 1086, 1126, 1217, 1235, 1332, 1419, 1476, 1485, 1504, 1582, 1599, 1608, 1612, 1617, 1618, 1619, 1693, 1754, 1776, 1899, 1903, 1960, 1980, 2004, 2020, 2068, 2151, 2174, 2179, 2184, 2198, 2245, 2336, 2339, 2364, 2367, 2374, 2378, 2394, 2423, 2435, 2513, 2518, 2715, 2739, 2844, 2899, 2906, 3062
<code>\GetTitleStringDisableCommands</code>	1815	<code>\ifZREF@found</code>	247, 2355, 2362
		<code>\ifZREF@immediate</code>	634, 700, 712, 716, 731
		<code>\ifZREF@pa@list</code>	1492, 1497
		<code>\ifZREF@pl@list</code>	1320, 1325
		<code>\ifzref@titleref@expand</code> ..	1784, 1800
		<code>\ifzref@titleref@stripperiod</code> ...	1797, 1808
		<code>\ifZREF@xr@toltxlabel</code> ...	2231, 2273
		<code>\ifZREF@xr@tozreflabel</code> ..	2217, 2259
		<code>\ifZREF@xr@urluse</code> ..	2101, 2380, 2410
		<code>\ifZREF@xr@verbose</code> ..	2219, 2261, 2282
		<code>\ifZREF@xr@zreflabel</code>	2032, 2118, 2132, 2173

<code>\immediate</code>	705, 1681	<code>\ltx@newif</code>	1320, 1492
<code>\in@</code>	314	<code>\ltx@one</code>	1393, 1548,
<code>\IncludeTests</code> ..	2920, 2955, 3009, 3033		1736, 2052, 2058, 2717, 2741, 2867
<code>\item</code>	107, 111, 113, 121, 125, 127	<code>\ltx@onelevel@sanitize</code>	557, 562
		<code>\ltx@ReturnAfterFi</code>	2341
		<code>\ltx@secondoftwo</code>	311,
			784, 797, 827, 886, 890, 1069, 2629
		<code>\ltx@space</code>	584, 586, 806, 815, 829,
			832, 1180, 1186, 1356, 1403,
			1591, 1619, 1662, 2530, 2537, 2600
		<code>\ltx@two</code>	2709, 2715, 2733, 2739
		<code>\ltx@undefined</code>	1249, 1431
		<code>\ltx@zero</code>	476, 1589,
			1642, 1712, 2048, 2477, 2541,
			2553, 2596, 2609, 2627, 2712, 2736
		<code>\luatexversion</code>	1243, 1425
K			
<code>\kern</code>	2887	M	
<code>\kv@define@key</code>	644, 655, 666, 671, 688	<code>\m@ne</code>	1053
<code>\kv@key</code> ...	692, 1611, 1613, 1614, 1628	<code>\M@sect</code>	1919
<code>\kv@parse</code>	689, 1607	<code>\M@TitleReference</code>	2299
<code>\kv@value</code>	690, 1608, 1609, 1616	<code>\mag</code>	1285, 2716, 2730, 2740, 2754
<code>\kvsetkeys</code>	633, 1832, 2045, 2863	<code>\mainmatter</code>	60, 134
		<code>\makeatletter</code>	11, 74,
			101, 2081, 2923, 2958, 2978, 3036
		<code>\makeatother</code>	16, 99, 2973
		<code>\makebox</code>	169, 170
		<code>\MakeRobustcommand</code>	232
		<code>\marginparsep</code>	1307
		<code>\marginparwidth</code>	1306
		<code>\mbox</code>	3044
		<code>\meaning</code> ..	2001, 2930, 2932, 2937, 2946
		<code>\mediaheight</code>	1291
		<code>\mediawidth</code>	1290
		<code>\MessageBreak</code> ..	270, 513, 564, 570,
			680, 1342, 1354, 1358, 1406,
			1517, 1540, 1543, 1567, 1569,
			1592, 1595, 1622, 1624, 1663,
			1664, 2063, 2064, 2113, 2130,
			2135, 2139, 2144, 2200, 2319,
			2328, 2441, 2454, 2601, 2613, 2614
L			
<code>\Label</code>	3038, 3056, 3058, 3061, 3068, 3073	N	
<code>\label</code>	969, 976, 1840, 2977	<code>\NeedsTeXFormat</code> ..	3, 188, 220, 960, 998,
<code>\leavevmode</code>	2866		1019, 1039, 1082, 1122, 1213,
<code>\List</code>	3037, 3040, 3082		1231, 1415, 1578, 1671, 1689,
<code>\LogTests</code>	2921, 2956, 3010, 3034		1772, 2016, 2419, 2431, 2509,
<code>\lst@caption</code>	1983		2840, 2895, 2915, 2951, 3005, 3029
<code>\lst@label</code>	1980	<code>\newcommand</code>	18,
<code>\lst@MakeCaption</code>	1979		78, 85, 91, 167, 968, 975, 981,
<code>\LTcc@ption</code>	1971		1136, 1153, 1154, 1223, 1586,
<code>\ltx@backslashchar</code>			2044, 2500, 2503, 2525, 2532,
	743, 1517, 1567, 2063		2539, 2551, 2608, 2863, 2959, 2965
<code>\ltx@car</code>	1611, 2154	<code>\newcount</code>	2047
<code>\ltx@carthree</code>	3050	<code>\newcounter</code>	6, 912, 1009, 1701
<code>\ltx@cdr</code>	1613, 2157	<code>\newif</code>	247, 700, 1784, 1797, 2032
<code>\ltx@empty</code>	290, 499,	<code>\newlabel</code>	2236, 2247, 2278
	565, 632, 719, 796, 1095, 1599,	<code>\newmarks</code>	1595
	1785, 1811, 1818, 1899, 1903,		
	1960, 1980, 2093, 2128, 2151,		
	2157, 2198, 2352, 2906, 2909, 2912		
<code>\ltx@firstofone</code>			
	254, 867, 878, 884, 1486, 1487		
<code>\ltx@firstoftwo</code>			
	799, 826, 827, 892, 1067, 2629, 2631		
<code>\ltx@firsttwo</code>	2620		
<code>\ltx@gobble</code>			
	250, 355, 390, 623, 663, 969,		
	970, 976, 1395, 1550, 1630, 1840		
<code>\ltx@gobblethree</code>	977		
<code>\ltx@gobbletwo</code> ..	694, 911, 1008, 1632		
<code>\ltx@ifpackageloaded</code>	2305		
<code>\ltx@ifUndefined</code>			
	229, 249, 257, 410, 877, 919,		
	1103, 1248, 1430, 1452, 1464,		
	1483, 1484, 1512, 1555, 1894,		
	1896, 2439, 2449, 2475, 2710, 2734		
<code>\ltx@ifundefined</code>			
	300, 485, 583, 760, 805, 940,		
	1255, 1274, 1327, 1437, 1499,		
	2030, 2290, 2291, 2564, 2635,		
	2636, 2637, 2650, 2651, 2652, 2662		
<code>\ltx@LocalAppendToMacro</code>			
	385, 403, 649, 659, 1565,		
	2209, 2249, 2252, 2357, 2360,		
	2369, 2376, 2382, 2389, 2396, 2408		

\newpage	143, 3081		
\nfss@text	928		
\nofiles	2916		
\NR@temp	1902, 1903		
\number	94, 109, 915, 920, 954, 1104, 1278, 1279, 1676, 1718, 1744, 1751, 2526, 2528, 2533, 2535, 2540, 2552, 2610, 2622, 2761, 2812, 2869, 3018, 3024		
\numexpr	94, 109, 115, 922, 1106, 1167, 1676, 1718, 2193, 2204, 2238, 2323, 2332, 2545, 2557, 2730, 2754, 2876		
O			
\oddsidemargin	1299		
\on@line	1678, 2128		
\openin	2110		
P			
\PackageError	258, 269, 294, 304, 489, 1590, 2440, 2452		
\PackageInfo	291, 520, 1679, 2117, 2129, 2220, 2262, 2283		
\PackageWarning	345, 363, 380, 398, 615, 679, 1620, 1661, 2112		
\PackageWarningNoLine	2199, 2318, 2327		
\page	2360, 2929, 2938, 2940, 2942		
\pagestyle	49		
\paperheight	1287		
\paperwidth	1286		
\par	3041		
\part	3062		
\pdf@escapehex	1489		
\pdf@strcmp	476		
\pdf@unescapehex	1490		
\pdfhorigin	1266, 1294		
\pdflastxpos	2462		
\pdflastypos	2463		
\pdfpageattr	1446, 1454, 1460		
\pdfpageheight	1265, 1293, 2661		
\pdfpagesattr	1447, 1466, 1472		
\pdfpagewidth	1264, 1292		
\pdfsavepos	2441, 2453, 2467, 2870, 2884		
\pdfTeXversion	2451		
\pdfvorigin	1267, 1295		
\ProcessOptions	217		
\protect	776		
\protected	238		
\protected@write	715		
\providecommand	281, 1674, 2029, 2860, 2861, 2862		
\ProvidesPackage	189, 221, 961, 999, 1020, 1040, 1083, 1123, 1214, 1232, 1416, 1579, 1672, 1690, 1773, 2017, 2420, 2432, 2510, 2841, 2896		
R			
\read	2163		
\refstepcounter	1031		
\renewcommand	7, 46, 914		
\RequirePackage	191, 196, 223, 224, 225, 226, 227, 230, 266, 271, 279, 963, 1001, 1006, 1022, 1042, 1043, 1044, 1085, 1090, 1091, 1125, 1130, 1131, 1132, 1133, 1216, 1221, 1222, 1234, 1239, 1240, 1241, 1418, 1423, 1459, 1471, 1479, 1480, 1481, 1581, 1692, 1697, 1775, 1780, 1781, 2019, 2024, 2025, 2422, 2434, 2446, 2512, 2517, 2522, 2524, 2843, 2848, 2850, 2898		
\reset@font	928		
\rightarrow	45		
\romannumeral	582, 804, 825, 1600, 2059, 2068, 2069		
S			
\section	63, 137, 145, 2976		
\setcounter	1011		
\setlength	2872		
\SetupKeyvalOptions	2033		
\space	778, 1454, 1466, 2131, 2132, 2136, 2140, 2145, 2441, 2453		
\spinemargin	1311		
\stepcounter	19, 1013, 1703, 1704		
\stockheight	1289		
\stockwidth	1288		
T			
\tableofcontents	59, 132		
\Test	2965, 2981, 2983, 2988, 2990, 2992, 2996, 2997, 3054, 3078		
\textbf	2976, 2986, 2999		
\textheight	1302		
\textit	2976, 2987, 2999		
\textwidth	1301		
\TeXeTstate	2477		
\the	13, 153, 155, 429, 444, 460, 556, 561, 620, 626, 742, 756, 922, 1015, 1059, 1100, 1106, 1167, 1326, 1346, 1353, 1354, 1355, 1357, 1402, 1404, 1405, 1460, 1472, 1498, 1522, 1538, 1539, 1541, 1794, 2187, 2193, 2204, 2238, 2323, 2332, 2462, 2463		
\thechapter	14		
\thefoo	7, 12, 20		
\theotype	2250		
\thepage	43, 44, 45, 713, 717, 778, 936, 1742		
\thezpage	18, 1742, 1748		
\thexref@unique	11, 914, 1739, 1740, 1747, 1748, 1750, 2871, 2873, 2876, 2877, 2885		

\title	2369, 2398	\zifrefundefined	9, 762
\toks@	423,	\zkvlabel	975
	429, 443, 444, 553, 556, 558,	\zlabel	12, 83, 104, 138, 146, 968
	561, 612, 619, 620, 626, 740,	\zlistpageattr	1493
	742, 755, 756, 1054, 1059, 1342,	\zlistpagelayout	17, 1321
	1346, 1352, 1353, 1517, 1522,	\zmakeperpage	18, 1710
	1537, 1538, 1788, 1794, 2172, 2187	\znextpage	15, 51, 54, 1150
\topmargin	1298	\znextpagesetup	15, 42, 1136
\TR@TitleReference ..	2294, 2354, 2393	\znonextpagename	46, 1153, 1201
\trimedge	1310	\zpageref	12, 126, 990
\trimtop	1313	\zposx	19, 153, 2500, 2876, 2877
\ttl@sect@i	1952	\zposy	19, 155, 2500
\ttl@straight@i	1958	\zref	12, 25, 26, 27, 28, 112,
			114, 123, 128, 129, 139, 981, 991
U			
\unexpanded	270, 275	\ZREF@@@delprop ...	434, 436, 471, 473
\UniqueCounterCall	1151	\ZREF@@@newprop	538, 542
\UniqueCounterNew	1134	\ZREF@@delprop	
\uppermargin	1314		424, 433, 447, 464, 470, 480
\url	2389	\ZREF@@extract	808, 814
\urluse	2384	\ZREF@@makeperpage ..	1712, 1718, 1722
\usepackage		\ZREF@@newprop	516, 527, 530, 534
	9, 41, 48, 70, 72, 2918, 2919,	\ZREF@@perpage@step	1727, 1735
	2953, 2954, 3007, 3008, 3031, 3032	\ZREF@abspos@body@x@center	2791
		\ZREF@abspos@body@x@left .	2789, 2816
		\ZREF@abspos@body@x@right	2790, 2813
		\ZREF@abspos@body@y@bottom	
			2796, 2808, 2830
		\ZREF@abspos@body@y@center	2800, 2831
		\ZREF@abspos@body@y@top	
			2792, 2797, 2801, 2829
		\ZREF@abspos@foot@x@center	2806
		\ZREF@abspos@foot@x@left	2804
		\ZREF@abspos@foot@x@right	2805
		\ZREF@abspos@foot@y@bottom	2807
		\ZREF@abspos@head@x@center	
			2771, 2791, 2806
		\ZREF@abspos@head@x@left	
			2757, 2768, 2772, 2789, 2804
		\ZREF@abspos@head@x@right	
			2767, 2790, 2805
		\ZREF@abspos@head@y@bottom	2781, 2793
		\ZREF@abspos@head@y@center	2785
		\ZREF@abspos@head@y@top	
			2775, 2782, 2786
		\ZREF@abspos@marginpar@x@center	2825
		\ZREF@abspos@marginpar@x@left ..	
			2811, 2822, 2826
		\ZREF@abspos@marginpar@x@right	2821
		\ZREF@abspos@marginpar@y@bottom	2830
		\ZREF@abspos@marginpar@y@center	2831
		\ZREF@abspos@marginpar@y@top ..	2829
		\ZREF@abspos@media@height	
			2649, 2683, 2689
		\ZREF@abspos@media@width	
			2634, 2676, 2680
		\ZREF@abspos@media@x@center ...	2678
		\ZREF@abspos@media@x@left	2672, 2679
		\ZREF@abspos@media@x@right	2675
V			
\value	13, 1100, 1326, 1498		
\verb	173		
\voffset	1297		
W			
\write	704, 705, 1681		
X			
\x	330,		
	335, 723, 728, 897, 899, 1193,		
	1209, 1275, 1282, 1345, 1348,		
	1351, 1391, 1520, 1525, 1536,		
	1546, 1644, 1649, 1652, 1658,		
	1717, 1720, 1998, 1999, 2004,		
	2171, 2174, 2179, 2184, 2194,		
	2197, 2207, 2208, 2239, 2244,		
	2970, 2971, 3049, 3050, 3077, 3078		
\XR@ext	2030		
Y			
\y	2000, 2001		
Z			
\z	2001, 2002, 2004		
\z@	2887		
\Z@D@page	1154		
\Z@L@LastPage	1056		
\Z@L@main	1055		
\Z@L@ZREF@temp	632, 636, 639, 650, 660		
\zdotfill	20, 170, 173, 2864		
\zdotfillsetup	20, 2863		
\zexternaldocument	20, 2075		
\ziflastpage	13, 1072		

<code>\ZREF@abspos@media@y@bottom</code> ...	2685	<code>\ZREF@df@min</code>	2855 , 2860 , 2878
<code>\ZREF@abspos@media@y@center</code> ...	2688	<code>\ZREF@df@unit</code>	2852 , 2861 , 2872
<code>\ZREF@abspos@media@y@top</code> .	2682 , 2700	<code>\ZREF@dotfill</code>	2874 , 2881 , 2890
<code>\ZREF@abspos@origin@x</code> ...	2709 , 2759	<code>\ZREF@ENV@line</code>	2909 , 2912
<code>\ZREF@abspos@origin@y</code> ...	2733 , 2777	<code>\zref@env@line</code>	2904 , 2905
<code>\ZREF@abspos@paper@x@center</code>		<code>\ZREF@extract</code>	803 , 820 , 823 , 875
.....	2697 , 2834	<code>\zref@extract</code>	8 ,
<code>\ZREF@abspos@paper@x@left</code>			95 , 96 , 109 , 140 , 803 , 823 , 852 ,
.....	2691 , 2758 , 2832		870 , 875 , 988 , 1109 , 1205 , 1357 ,
<code>\ZREF@abspos@paper@x@right</code> 2694 , 2833			1404 , 1405 , 1529 , 1559 , 1747 ,
<code>\ZREF@abspos@paper@y@bottom</code>			1748 , 1841 , 2501 , 2504 , 2676 ,
.....	2701 , 2836		2680 , 2683 , 2689 , 2695 , 2698 ,
<code>\ZREF@abspos@paper@y@center</code>			2703 , 2707 , 2722 , 2746 , 2760 ,
.....	2705 , 2837		2762 , 2764 , 2769 , 2773 , 2778 ,
<code>\ZREF@abspos@paper@y@top</code>			2779 , 2783 , 2787 , 2794 , 2798 ,
....	2700 , 2702 , 2706 , 2776 , 2835		2802 , 2809 , 2814 , 2817 , 2818 ,
<code>\ZREF@abspos@stock@x@center</code> ...	2834		2823 , 2827 , 2992 , 2996 , 2997 , 3049
<code>\ZREF@abspos@stock@x@left</code>	2832	<code>\ZREF@extractdefault</code> 824 , 840 , 843 , 874	
<code>\ZREF@abspos@stock@x@right</code>	2833	<code>\zref@extractdefault</code>	
<code>\ZREF@abspos@stock@y@bottom</code> ...	2836	8 , 116 , 117 , 816 , 843 ,
<code>\ZREF@abspos@stock@y@center</code> ...	2837		863 , 869 , 874 , 1065 , 1066 , 1163 ,
<code>\ZREF@abspos@stock@y@top</code>	2835		1178 , 1224 , 1750 , 2292 , 2295 ,
<code>\ZREF@abspos@used</code>	2586 , 2588		2296 , 2300 , 2301 , 2304 , 2306 ,
<code>\ZREF@absposnum</code>	2545 , 2557 , 2563		2307 , 2309 , 2311 , 2528 , 2535 ,
<code>\zref@absposnumused</code>	2608		2594 , 2622 , 2761 , 2812 , 2988 , 2990
<code>\zref@absposnumx</code> ...	2527 , 2539 , 2577	<code>\ZREF@false</code>	676 , 686
<code>\zref@absposnumy</code> ...	2534 , 2551 , 2580	<code>\ZREF@foundfalse</code>	2353
<code>\zref@absposused</code>	2585	<code>\ZREF@foundtrue</code>	2400
<code>\zref@absposx</code>	2525 , 2571	<code>\ZREF@getcurrent</code> ..	581 , 592 , 595 , 873
<code>\zref@absposy</code>	2532 , 2574	<code>\zref@getcurrent</code>	
<code>\zref@addprop</code> ...	6 , 76 , 359 , 1016 ,	7 , 595 , 868 , 873 , 2981 , 2983
	1028 , 1093 , 1096 , 1280 , 1461 ,	<code>\zref@hex</code>	1460 , 1472 , 1486 , 1489
	1473 , 1628 , 1783 , 2428 , 2523 , 2944	<code>\zref@ifabsposnumundefined</code>	2621 , 2626
<code>\zref@addprops</code>		<code>\zref@ifabsposundefined</code>	2619
..	6 , 15 , 340 , 937 , 1319 , 1700 , 2464	<code>\ZREF@ifDefinable</code>	241 ,
<code>\ZREF@addtoks</code>	754		762 , 990 , 993 , 1072 , 1114 , 1150 ,
<code>\ZREF@base@ok</code>	957 , 964 , 1002 ,		1321 , 1493 , 1710 , 1764 , 1831 ,
	1023 , 1045 , 1086 , 1126 , 1217 ,		1834 , 2075 , 2485 , 2490 , 2495 , 2864
	1235 , 1419 , 1582 , 1693 , 1776 ,	<code>\ZREF@iflastpage</code> ...	1073 , 1075 , 1075
	2020 , 2423 , 2435 , 2513 , 2844 , 2899	<code>\zref@iflastpage</code>	13 , 1064 , 1078
<code>\ZREF@call</code> 1159 , 1174 , 1183 , 1187 , 1195		<code>\zref@iflistcontainsprop</code>	7 ,
<code>\ZREF@def@abspos</code>			309 , 344 , 362 , 379 , 397 , 647 , 657
....	2571 , 2574 , 2577 , 2580 , 2582	<code>\zref@iflistundefined</code>	
<code>\zref@def@absposnumx</code>	2576	7 , 288 , 299 , 303 , 310
<code>\ZREF@def@absposnumy</code>	2582	<code>\zref@ifpropundefined</code> ..	7 , 484 , 488 ,
<code>\zref@def@absposnumy</code>	2579		518 , 547 , 614 , 827 , 1400 , 1641 , 2347
<code>\zref@def@absposx</code>	2570	<code>\ZREF@ifrefcontainsprop</code> ...	786 , 794
<code>\zref@def@absposy</code>	2573	<code>\zref@ifrefcontainsprop</code>	
<code>\ZREF@def@extract</code>	845 , 847	...	9 , 782 , 1402 , 2308 , 2404 , 2405
<code>\zref@def@extract</code>	8 , 844	<code>\ZREF@ifrefundefined</code>	
<code>\ZREF@def@extractdefault</code> ...	856 , 858	764 , 767 , 1160 , 1171 , 1181
<code>\zref@def@extractdefault</code>	855	<code>\zref@ifrefundefined</code>	9 ,
<code>\ZREF@default</code>	561 , 562 , 571		759 , 769 , 775 , 783 , 826 , 1172 ,
<code>\zref@default</code> ...	9 , 530 , 806 , 925 , 927		1355 , 1541 , 1740 , 2542 , 2554 ,
<code>\ZREF@delprop</code>			2590 , 2620 , 2628 , 2873 , 2994 , 3046
.....	412 , 415 , 417 , 452 , 455 , 457	<code>\ZREF@immediatefalse</code>	677
<code>\zref@delprop</code>	411 , 451	<code>\ZREF@immediatetrue</code>	674 , 703
<code>\ZREF@df@dot</code>	2858 , 2862 , 2891	<code>\ZREF@label</code> 602 , 626 , 636 , 639 , 709 , 1059	

<code>\zref@label</code>	7, 596, 972, 3039	<code>\ZREF@org@@opargbegintheorem</code> ..	1993
<code>\zref@labelbykv</code>	629, 979	<code>\ZREF@org@@part</code>	1854
<code>\zref@labelbylist</code>	8, 597, 599, 1100, 1739, 2487	<code>\ZREF@org@@schapter</code>	1878
<code>\zref@labelbyprops</code>	8, 88, 609, 1158, 2492, 2497, 2871, 2885	<code>\ZREF@org@@sect</code>	1866
<code>\zref@listexists</code>	6, 302, 321, 341, 360, 376, 395, 418, 458, 601	<code>\ZREF@org@@spart</code>	1872
<code>\zref@listforloop</code>	320, 656	<code>\ZREF@org@@ssect</code>	1884
<code>\zref@listpageattr</code>	1493	<code>\ZREF@org@beamer@section</code>	1934
<code>\zref@listpagelayout</code>	1321	<code>\ZREF@org@beamer@subsection</code> ...	1940
<code>\zref@localaddprop</code> ...	394, 2936, 2939	<code>\ZREF@org@beamer@subsubsection</code>	1946
<code>\zref@localaddprops</code>	375	<code>\ZREF@org@descriptionlabel</code>	1890
<code>\zref@localdelprop</code>	414, 454, 668	<code>\ZREF@org@lst@MakeCaption</code>	1986
<code>\ZREF@mainlist</code>	597, 931, 934, 937, 1016, 1028, 1783, 2428	<code>\ZREF@org@LT@cc@ption</code>	1972
<code>\ZREF@makeperpage@opt</code> ...	1712, 1715	<code>\ZREF@org@M@sect</code>	1925
<code>\ZREF@MARKS@DefineProp</code>	1604, 1605, 1606, 1640	<code>\ZREF@org@refstepcounter</code>	1033
<code>\zref@marks@register</code>	1586, 1591, 1623, 1662	<code>\ZREF@org@stepcounter</code> ...	1703, 1708
<code>\ZREF@name</code>	228, 258, 269, 291, 294, 304, 345, 363, 380, 398, 489, 508, 520, 524, 548, 563, 615, 679, 691, 1590, 2440, 2452	<code>\ZREF@org@testdef</code>	1328, 1330, 1500, 1502
<code>\ZREF@NAME@bot</code>	1619, 1639	<code>\ZREF@org@thepage</code>	713, 717
<code>\ZREF@NAME@first</code>	1618, 1638	<code>\ZREF@org@ttl@sect@i</code>	1954
<code>\ZREF@NAME@top</code>	1617, 1637	<code>\ZREF@org@ttl@straight@i</code>	1965
<code>\zref@newlabel</code>	8, 281, 284, 749, 2191, 2277	<code>\ZREF@org@write</code>	704, 705
<code>\zref@newlist</code>	6, 287, 934, 1049, 1092, 1614, 1699, 2461, 2933	<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@newprop</code>	497, 500, 503	<code>\ZREF@pa@AfterLastShipout</code>	1496, 1574
<code>\zref@newprop</code>	7, 12, 13, 14, 75, 494, 935, 936, 939, 946, 950, 954, 1015, 1027, 1277, 1316, 1317, 1318, 1460, 1472, 1645, 1653, 2026, 2027, 2028, 2348, 2462, 2463, 2903, 2904, 2931, 2979	<code>\ZREF@pa@AtVeryEnd</code> ..	1509, 1512, 1565
<code>\ZREF@NewPropAnchor</code> ..	938, 2077, 2427	<code>\ZREF@pa@ListPage</code>	1519, 1535
<code>\ZREF@NewPropPageValue</code>	953, 1094, 1698	<code>\ZREF@pa@listtrue</code>	1494
<code>\ZREF@NewPropTheotype</code>	949, 2248	<code>\ZREF@page@max</code> ..	1326, 1392, 1498, 1547
<code>\ZREF@NewPropTitle</code> ...	944, 1782, 2078	<code>\zref@pageattr</code>	1527
<code>\ZREF@nextpage</code>	1151, 1155	<code>\zref@pageattr@used</code>	1532
<code>\ZREF@nil</code>	544, 796, 835, 2164, 2170, 2176, 2181, 2191, 2207, 2236, 2244, 2335, 2342, 2351, 2354, 2393	<code>\ZREF@pagenum@last</code>	1177, 1180
<code>\ZREF@NOVALUE</code>	802	<code>\ZREF@pagenum@this</code>	1162, 1167, 1170, 1180, 1186
<code>\ZREF@novalue</code>	795, 796, 802	<code>\ZREF@par</code>	507, 532
<code>\ZREF@np@call@next</code> ..	1145, 1149, 1204	<code>\ZREF@param</code>	421, 422, 441, 459, 476, 645, 646, 647, 651, 672, 673, 676, 681
<code>\ZREF@np@call@nonext</code>	1142, 1148, 1200	<code>\ZREF@patch</code> ..	248, 1030, 1845, 1851, 1857, 1863, 1869, 1875, 1881, 1887, 1918, 1931, 1937, 1943, 1951, 1957, 1970, 1978, 1990, 2005
<code>\ZREF@np@call@unknown</code>	1138, 1147, 1196	<code>\zref@pdfpageattr</code>	1456, 1542
<code>\ZREF@np@setup@i</code>	1137, 1140	<code>\zref@pdfpageattr@used</code>	1457
<code>\ZREF@np@setup@ii</code>	1141, 1144	<code>\zref@pdfpagesattr</code> ..	1468, 1557, 1568
<code>\ZREF@number</code>	919, 1529, 1533, 1588, 2627	<code>\zref@pdfpagesattr@used</code> ..	1469, 1562
<code>\ZREF@org@@begintheorem</code>	2008	<code>\ZREF@pl@AfterLastShipout</code>	1324, 1411
<code>\ZREF@org@@caption</code>	1848	<code>\ZREF@pl@AtVeryEnd</code>	1337, 1340
<code>\ZREF@org@@chapter</code>	1860, 1916	<code>\ZREF@pl@ListEntry</code>	1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1399
		<code>\ZREF@pl@ListPage</code>	1344, 1350
		<code>\ZREF@pl@listtrue</code>	1322

\zref@pos@label@used	2600	\ZREF@TempNum	1588, 1589, 1593, 1600, 1642, 1655
\zref@pos@num@used	2614	\zref@thepage	14, 1108, 1117
\zref@prop 323, 331, 332, 336, 657, 661		\zref@thepage@atbegshi@hook	1095, 1099
\zref@propexists 7, 343, 361,		\zref@thepage@name	14, 1103, 1109, 1112, 1166
378, 396, 487, 577, 646, 667, 982		\zref@thepage@refused ...	1111, 1116
\ZREF@refname@next	1165, 1172, 1181, 1205	\ZREF@titleref	1835, 1837
\ZREF@refname@this	1157, 1158, 1160, 1163	\zref@titleref@cleanup ..	1786, 1826
\ZREF@RefPrefix . 283, 285, 1332, 1504		\zref@titleref@current	944, 1805, 1809, 1810, 1829
\ZREF@refused	772, 774	\ZREF@titleref@hook	1785, 1789, 1793, 1816
\zref@refused	8, 768, 771, 848, 859, 987, 994,	\zref@titleref@setcurrent	1799, 1847, 1853, 1859, 1865,
1076, 1077, 1112, 1227, 1533,		1871, 1877, 1883, 1889, 1897,	
1563, 1839, 2589, 2597, 2610, 3045		1900, 1904, 1908, 1910, 1921,	
\zref@require@unique	11, 908, 1702, 2849	1923, 1933, 1939, 1945, 1953,	
\ZREF@Robust	231, 237, 243, 284, 287, 302, 309,	1961, 1963, 1973, 1982, 1992, 2007	
340, 359, 375, 394, 411, 414,		\zref@titleref@stripperperiodtrue	1798
451, 454, 487, 494, 546, 576,		\ZREF@true	673, 687
596, 599, 609, 629, 701, 771,		\ZREF@u@getcurrent	590
844, 855, 866, 882, 908, 924,		\zref@unhex 1487, 1490, 1528, 1558	
930, 1111, 1532, 1562, 1786,		\ZREF@UpdatePdfTeX ...	246, 2443, 2456
1799, 2570, 2573, 2576, 2579, 2585		\ZREF@value	556, 557, 570
\ZREF@SavedEscapechar	460, 467	\ZREF@wrapper@babel	899, 905
\zref@savepos ... 20, 2465, 2473, 2479		\zref@wrapper@babel	11, 140, 764, 772, 845,
\ZREF@savepos@ok	2506, 2518	856, 882, 972, 979, 983, 1073,	
\zref@setcurrent	7, 81, 540, 576, 696, 1032	1835, 2571, 2574, 2577, 2580, 2586	
\zref@setdefault	9, 924, 927	\zref@wrapper@immediate	11, 87, 635, 701, 1058
\zref@setmainlist	9, 930	\ZREF@wrapper@unexpanded ...	866, 880
\zref@showprop	546	\zref@wrapper@unexpanded	11, 867, 872, 877, 2287, 2967
\ZREF@STAR	1612, 1636	\ZREF@wu@extract	818, 870
\ZREF@stripperperiod	1810, 1818	\ZREF@wu@extractdefault ...	838, 869
\ZREF@temp	193, 200, 201, 202, 203, 204,	\ZREF@wu@getcurrent	590, 868
205, 206, 207, 208, 209, 210,		\ZREF@X	496, 499, 536
211, 212, 213, 214, 215, 216,		\zref@xr@	2041
232, 233, 439, 440, 441, 719,		\ZREF@xr@@AddUrl	2053, 2056
740, 741, 749, 1247, 1264, 1265,		\ZREF@xr@input	2184, 2279
1266, 1267, 1271, 1285, 1286,		\ZREF@xr@AddURL	2049, 2102, 2381
1287, 1288, 1289, 1290, 1291,		\ZREF@xr@checkfile ..	2106, 2109, 2159
1292, 1293, 1294, 1295, 1296,		\ZREF@xr@checkkey	2337, 2346
1297, 1298, 1299, 1300, 1301,		\ZREF@xr@checklist	2207, 2335
1302, 1303, 1304, 1305, 1306,		\zref@xr@ext	21, 2029, 2097
1307, 1308, 1309, 1310, 1311,		\ZREF@xr@externaldocument	2084, 2087, 2091
1312, 1313, 1314, 1315, 1331,		\ZREF@xr@externalfile	2094, 2095, 2213, 2255
1332, 1429, 1446, 1447, 1451,		\ZREF@xr@file ...	2095, 2110, 2113,
1462, 1474, 1476, 1482, 1483,		2119, 2130, 2153, 2201, 2320, 2329	
1484, 1485, 1503, 1504, 1611,		\ZREF@xr@filelist	2093,
1612, 1959, 1960, 2051, 2061,		2151, 2154, 2156, 2157, 2185, 2186	
2064, 2068, 2593, 2596, 2597,		\ZREF@xr@found .	2121, 2131, 2193, 2238
2709, 2712, 2715, 2717, 2720,		\ZREF@xr@graburl	2097, 2099
2733, 2736, 2739, 2741, 2744, 2869			
\ZREF@TempName .. 1587, 1599, 1600,			
1602, 1628, 1641, 1645, 1653, 1664			

\ZREF@xr@ignored@empty	2226, 2241, 2243, 2250, 2253, 2268
.... 2122, 2134, 2136, 2203, 2204	\ZREF@xr@theURL
\ZREF@xr@ignored@ltx	2059, 2061, 2063, 2069, 2104, 2385
.... 2124, 2143, 2145, 2331, 2332	\ZREF@xr@tolabel ... 2232, 2274, 2281
\ZREF@xr@ignored@zref	\ZREF@xr@URL ... 2047, 2057, 2058, 2059
.... 2123, 2138, 2140, 2322, 2323	\ZREF@xr@url ... 2100, 2102, 2103, 2411
\ZREF@xr@line .. 2163, 2164, 2176, 2181	\ZREF@xr@urlcheck .. 2216, 2258, 2403
\ZREF@xr@list	\ZREF@xr@zref@ignorewarning
2197, 2198	2228, 2270, 2317
\ZREF@xr@ltx@ignorewarning 2326	\ZREF@xr@zref@newlabel .. 2174, 2277
\ZREF@xr@newlabel	2179, 2278
\ZREF@xr@prefix	2092, 2192,
2228, 2232, 2237, 2263, 2270, 2274	\ZREF@xr@zref@labelfalse
\ZREF@xr@process@label .. 2181, 2236	2083
\ZREF@xr@process@zref@label 2176, 2191	\ZREF@xr@zref@labeltrue
\ZREF@xr@process@file 2109, 2162	2086
\ZREF@xr@process@line 2164, 2170	\ZREF@zref
\ZREF@xr@refname	983, 986
2192, 2218, 2225, 2237, 2260, 2267	\ZREF@zsavepos . 2470, 2487, 2492, 2497
\ZREF@xr@relax	\zrefused 12, 92, 93, 161, 162, 163, 993
2280, 2367	\zruns
\ZREF@xr@scanparams	17, 1674, 3013, 3018, 3024
2242, 2351	\zsavepos
\ZREF@xr@scantitleref ... 2354, 2393	19, 157, 158, 2485
\ZREF@xr@temp	\zsaveposx
2366, 2367	19, 2490
\ZREF@xr@tempname 2195, 2196, 2216,	\zsaveposy
2221, 2232, 2240, 2241, 2258, 2274	2495
\ZREF@xr@temprefname	\zthepage
2196, 2208, 2210,	14, 1114
	\ztitleref
	18, 1834, 3042
	\ztitlerefsetup
	19, 1819
	\ztotpages
	16, 124, 1223
	\zunknownnextpagename . 15, 1154, 1197
	\zunmakeperpage
	18, 1764
	\zxrsetup
	21, 2044