

The zref package

Heiko Oberdiek*

<heiko.oberdiek at gmail.com>

2016/05/21 v2.26

Abstract

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

Contents

1	Introduction	4
1.1	Standard \LaTeX behaviour	4
1.2	Basic idea	4
1.3	Interfaces	5
2	Interface for programmers	5
2.1	Entities	5
2.2	Property list	6
2.3	Property	6
2.4	Reference generation	7
2.5	Data extraction	7
2.6	Setup	8
2.7	Declared properties	9
2.8	Wrapper for advanced situations	10
2.9	Counter for unique names	10
3	User interface	10
3.1	Module <code>user</code>	10
3.2	Module <code>abspage</code>	11
3.3	Module <code>lastpage</code>	11
3.3.1	Tests for last page	12
3.3.2	Example	12
3.4	Module <code>thepage</code>	13
3.5	Module <code>nextpage</code>	13
3.5.1	Configuration	13
3.5.2	Example	14
3.6	Module <code>totpages</code>	14
3.7	Module <code>pagelayout</code>	15
3.8	Module <code>marks</code>	15
3.9	Module <code>runs</code>	16
3.10	Module <code>perpage</code>	16
3.11	Module <code>counter</code>	16
3.12	Module <code>titleref</code>	17

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

3.13	Module <code>savepos</code>	17
3.14	Module <code>dotfill</code>	18
3.15	Module <code>env</code>	18
3.16	Module <code>xr</code>	18
4	ToDo	19
5	Example	19
6	Implementation	22
6.1	Package <code>zref</code>	22
6.1.1	Identification	22
6.1.2	Load basic module	22
6.1.3	Process options	22
6.2	Module <code>base</code>	22
6.2.1	Prefixes	22
6.2.2	Identification	23
6.2.3	Utilities	23
6.2.4	Check for ε -TeX	24
6.2.5	Auxiliary file stuff	24
6.2.6	Property lists	25
6.2.7	Properties	29
6.2.8	Reference generation	31
6.2.9	Reference querying and extracting	34
6.2.10	Compatibility with <code>babel</code>	37
6.2.11	Unique counter support	38
6.2.12	Utilities	38
6.2.13	Setup	38
6.3	Module <code>user</code>	39
6.4	Module <code>abspage</code>	40
6.5	Module <code>counter</code>	41
6.6	Module <code>lastpage</code>	41
6.7	Module <code>thepage</code>	42
6.8	Module <code>nextpage</code>	43
6.9	Module <code>totpages</code>	45
6.10	Module <code>pagelayout</code>	45
6.10.1	Support for LuaTeX	45
6.10.2	Define layout properties	46
6.11	Module <code>pageattr</code>	48
6.12	Module <code>marks</code>	51
6.13	Module <code>runs</code>	53
6.14	Module <code>perpage</code>	53
6.15	Module <code>titleref</code>	55
6.15.1	Implementation	55
6.15.2	User interface	57
6.15.3	Patches for section and caption commands	57
6.15.4	Environment <code>description</code>	58
6.15.5	Class <code>memoir</code>	58
6.15.6	Class <code>beamer</code>	59
6.15.7	Package <code>titlesec</code>	59
6.15.8	Package <code>longtable</code>	60
6.15.9	Package <code>listings</code>	60
6.15.10	Theorems	60
6.16	Module <code>xr</code>	60
6.17	Module <code>hyperref</code>	68
6.18	Module <code>savepos</code>	68
6.18.1	Identification	68
6.18.2	Availability	69

6.18.3	Setup	69
6.18.4	User macros	69
6.19	Module <code>abspos</code>	70
6.19.1	Identification	70
6.19.2	Media	73
6.19.3	Paper	74
6.19.4	Origin	75
6.19.5	Header	75
6.19.6	Body	76
6.19.7	Footer	77
6.19.8	Marginal notes	77
6.19.9	Stock paper	78
6.20	Module <code>dotfill</code>	78
6.21	Module <code>env</code>	79
7	Test	80
7.1	<code>\zref@localaddprop</code>	80
7.2	Module <code>base</code>	80
7.3	Module <code>runs</code>	81
7.4	Module <code>titleref</code>	81
8	Installation	83
8.1	Download	83
8.2	Bundle installation	83
8.3	Package installation	83
8.4	Refresh file name databases	84
8.5	Some details for the interested	84
9	Catalogue	85
10	References	85
11	History	86
[2006/02/20	v1.0]	86
[2006/05/03	v1.1]	86
[2006/05/25	v1.2]	86
[2006/09/08	v1.3]	86
[2007/01/23	v1.4]	86
[2007/02/18	v1.5]	86
[2007/04/06	v1.6]	86
[2007/04/17	v1.7]	86
[2007/04/22	v1.8]	86
[2007/05/02	v1.9]	86
[2007/05/06	v2.0]	86
[2007/05/28	v2.1]	87
[2008/09/21	v2.2]	87
[2008/10/01	v2.3]	87
[2009/08/07	v2.4]	87
[2009/12/06	v2.5]	87
[2009/12/07	v2.6]	87
[2009/12/08	v2.7]	87
[2010/03/26	v2.8]	87
[2010/03/29	v2.9]	87
[2010/04/08	v2.10]	87
[2010/04/15	v2.11]	88
[2010/04/17	v2.12]	88
[2010/04/19	v2.13]	88
[2010/04/22	v2.14]	88

[2010/04/23 v2.15]	88
[2010/04/28 v2.16]	88
[2010/05/01 v2.17]	89
[2010/05/13 v2.18]	89
[2010/10/22 v2.19]	89
[2011/02/12 v2.20]	89
[2011/03/18 v2.21]	89
[2011/10/05 v2.22]	89
[2011/12/05 v2.23]	89
[2012/04/04 v2.24]	89
[2016/05/16 v2.25]	89
[2016/05/21 v2.26]	90

12 Index **91**

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@ifistundefinedexp {<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 LaTeX 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 `\zref@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	<empty>
	page	main	<empty>
abspage	abspage	main	0
counter	counter	main	<empty>
hyperref	anchor	main	<empty>
	url		<empty>
pageattr	pdfpageattr	thepage	...
	pdfpagesattr	LastPage	...
pagelayout ¹	mag	thepage	\number\mag
	paperwidth	thepage	\number\paperwidth
	paperheight	thepage	\number\paperheight
	stockwidth	thepage	\number\stockwidth
	stockheight	thepage	\number\stockheight
	pdfpageheight	thepage	\number\pdfpageheight
	pdfpagewidth	thepage	\number\pdfpagewidth
	pdfhorigin	thepage	\number\pdfhorigin
	pdfvorigin	thepage	\number\pdfvorigin
	hoffset	thepage	\number\hoffset
	voffset	thepage	\number\voffset
	topmargin	thepage	\number\topmargin
	oddsidemargin	thepage	\number\oddsidemargin
	evensidemargin	thepage	\number\evensidemargin
	textwidth	thepage	\number\textwidth
	textheight	thepage	\number\textheight
	headheight	thepage	\number\headheight
	headsep	thepage	\number\headsep
	footskip	thepage	\number\footskip
marginparwidth	thepage	\number\marginparwidth	
marginparsep	thepage	\number\marginparsep	
columnwidth	thepage	\number\columnwidth	
columnsep	thepage	\number\columnsep	
perpage	pagevalue	perpage	0
	page	perpage	<empty>
	abspage	perpage	0
savepos	posx	savepos	0
	posy	savepos	0
titleref	title	main	<empty>
xr	anchor		<empty>
	externaldocument		<empty>
	thetype		<empty>
	title		<empty>
	url		<empty>

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

2.8 Wrapper for advanced situations

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

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

`\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:

- babel Babel shorthands are allowed.
- robust Robust macro.
- exp Expandable version:
 - robust, unless the extracted values are fragile,
 - no babel shorthand support.
- exp2 Expandable like `exp` and:
 - expandable in exact two steps.

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

`\zlabel {⟨refname⟩}babel`

Similar to `\label`. It generates a label with name `⟨refname⟩` in the new reference scheme.

`\zref [⟨propname⟩] {⟨refname⟩}babel`

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

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

`\zpageref {⟨refname⟩}babel`

Convenience macro, similar to `\pageref`.

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

`\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, `\zrefused` 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:

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

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}[2016/05/21]
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.

<code>\zunknownnextpagename</code> <code>\znonextpagename</code>

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 exist. That means that the current page is last page. The default is empty.

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

According to the case (see `\znextpage`) macro `\znextpage` calls an internal macro with an argument. The argument is either `\thepage` of the next page or one of `\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}[2016/05/21]
42 \znextpagesetup
43 {\thepage}% next page is unknown
44 {\thepage\ (#1)}% this page is last page
45 {\thepage\ $\rightarrow$ #1}% next page is known
46 \renewcommand*{\znonextpagename}{last page}
47
48 \usepackage{fancyhdr}
49 \pagestyle{fancy}
50 \fancyhf{}
51 \fancyhead[LE,RO]{\znextpage}
52 \fancypagestyle{plain}{%
53 \fancyhf{}%
54 \fancyhead[LE,RO]{\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 `TEX` engine like `pdfTEX`. 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>	<code>mag</code>	
<code>\paperwidth</code>	<code>paperwidth</code>	
<code>\paperheight</code>	<code>paperheight</code>	
<code>\stockwidth</code>	<code>stockwidth</code>	class <code>memoir</code>
<code>\stockheight</code>	<code>stockheight</code>	class <code>memoir</code>
<code>\pdfpagewidth</code>	<code>pdfpagewidth</code>	<code>pdfT_EX</code> , <code>LuaT_EX</code>
<code>\pdfpageheight</code>	<code>pdfpageheight</code>	<code>pdfT_EX</code> , <code>LuaT_EX</code>
<code>\pdfhorigin</code>	<code>pdfhorigin</code>	<code>pdfT_EX</code> , <code>LuaT_EX</code>
<code>\pdfvorigin</code>	<code>pdfvorigin</code>	<code>pdfT_EX</code> , <code>LuaT_EX</code>
<code>\hoffset</code>	<code>hoffset</code>	
<code>\voffset</code>	<code>voffset</code>	
<code>\topmargin</code>	<code>topmargin</code>	
<code>\oddsidemargin</code>	<code>oddsidemargin</code>	
<code>\evensidemargin</code>	<code>evensidemargin</code>	
<code>\textwidth</code>	<code>textwidth</code>	
<code>\textheight</code>	<code>textheight</code>	
<code>\headheight</code>	<code>headheight</code>	
<code>\headsep</code>	<code>headsep</code>	
<code>\footskip</code>	<code>footskip</code>	
<code>\marginparwidth</code>	<code>marginparwidth</code>	
<code>\marginparsep</code>	<code>marginparsep</code>	
<code>\columnwidth</code>	<code>columnwidth</code>	
<code>\columnsep</code>	<code>columnsep</code>	

`\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 \LaTeX runs since last `.aux` file creation and prints the number in the `.log` file.

`\zrunsexp`

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

3.10 Module perpage

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

`\zmakeperpage [reset] {counter}`

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

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

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

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

`\thezpage`
counter `zpage`

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

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

`\zunmakeperpage {counter}`

The reset mechanism for this counter is deactivated.

3.11 Module counter

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

3.12 Module `titleref`

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

`\ztitleref {⟨refname⟩}^babel`

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

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

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

`title=⟨value⟩`
Sets the current title.

`stripperperiod=true|false`
Follow package `nameref` that removes a last period. Default: `true`.

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

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

3.13 Module `savepos`

This option supports a feature that `pdfTeX` provides (and `XYTeX`). `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 pdf_TE_X'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.

totxlabel: Boolean option. The found references are imported as `LATEX` labels. Packages `nameref`, `titleref` and class `memoir` are supported.

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

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

`\zref@xr@ext`

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

4 ToDo

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

- Other applications: `autoref`, `hyperref`, ...

5 Example

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

```

128 The title of chapter \zref{chap:next} %
129 is ``\zref[chaptitle]{chap:next}``.
130 \end{itemize}
131
132 \tableofcontents
133
134 \mainmatter
135 \ChapterStart{first}{First chapter}
136

```

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

```

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

```

Here an example follows that makes use of pdf \TeX '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 \LaTeX compile run.

```

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

```

With `\zrefused` \LaTeX is notified, if the references are not yet available and \LaTeX 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*{\dftest}[1]{%
168 #1&
169 [\makebox[#{#1}]{\dotfill}]&
170 [\makebox[#{#1}]{\zdotfill}]\
171 }
172 \begin{tabular}{rll}
173 & [\verb|\dotfill|] & [\verb|\zdotfill|]\
174 \dftest{0.43em}
175 \dftest{0.44em}
176 \dftest{0.45em}
177 \dftest{0.87em}
178 \dftest{0.88em}

```

```

179 \dftest{0.89em}
180 \dftest{1.31em}
181 \dftest{1.32em}
182 \dftest{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 [2016/05/21 v2.26 A new reference scheme for LaTeX (HO)]%

```

6.1.2 Load basic module

```

191 \RequirePackage{zref-base}[2016/05/21]

```

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}[2016/05/21]%
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 $\langle listname \rangle$.

\Z@D@propname: The default value for property $\langle propname \rangle$.

\Z@E@propname: Extract function for property $\langle propname \rangle$.

\Z@X@propname: Information whether a property value for property $\langle propname \rangle$ is expanded immediately or at shipout time.

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

\Z@R@labelname: Data for reference $\langle labelname \rangle$.

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

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

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

ZREF@TR: Setup for module `titleref`.

6.2.2 Identification

```

219  $\langle *base \rangle$ 
220 \NeedsTeXFormat{LaTeX2e}
221 \ProvidesPackage{zref-base}%
222 [2016/05/21 v2.26 Module base for zref (HO)]%
```

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@stricmp}{%

```

\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@stricmp{#1}{\ZREF@param}=\ltx@zero
477 \else
478 \expandafter\noexpand\csname#1\endcsname
479 \fi
480 \expandafter\ZREF@@delprop

```

```

481 \fi
482 }%

483 }

```

6.2.7 Properties

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

```

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

```

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

```

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

```

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

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

`\Z@E@propname`: Extract function.

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

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

```

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

```

`\ZREF@newprop`

```

503 \def\ZREF@newprop#1{%
504 \edef\ZREF@P{#1}%
505 \@onelevel@sanitize\ZREF@P
506 \begingroup
507 \ifx\ZREF@P\ZREF@par
508 \@PackageError\ZREF@name{%
509 Invalid property name `\'ZREF@P'%
510 }{%
511 The property name `par' is not allowed %
512 because of internal reasons.%
513 \MessageBreak
514 \@ehc
515 }%
516 \def\ZREF@@@newprop[##1]##2{\endgroup}%
517 \else

```

```

518 \zref@ifpropundefined\ZREF@P{%
519 \endgroup
520 \PackageInfo\ZREF@name{%
521 New property: \ZREF@P
522 }%
523 }{%
524 \@PackageError\ZREF@name{%
525 Property ``\ZREF@P' already exists%
526 }\@ehc
527 \def\ZREF@@newprop[##1]##2{\endgroup}%
528 }%
529 \fi
530 \ifnextchar[\ZREF@@newprop{\ZREF@@newprop[\zref@default]}%
531 }

```

\ZREF@par

```

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

```

\ZREF@@newprop

```

534 \def\ZREF@@newprop[#1]{%
535 \global\@namedef{Z@D@\ZREF@P}#{1}%
536 \global\expandafter\let\csname Z@X@\ZREF@P\endcsname\ZREF@X
537 \begingroup\expandafter\endgroup
538 \expandafter\ZREF@@@newprop\csname\ZREF@P\endcsname
539 \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@gobbetwo
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 `33is` 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 \@for\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 pdfT_EX that allows the detection of contexts. Then the shorthand can detect, if they are executed inside `\csname` and protect themselves automatically.

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

```

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 LATEX and a warning is given.
771 \ZREF@Robust\def\zref@refused#1{%
772   \zref@wrapper@babel\ZREF@refused{#1}%
773 }

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

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

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

```

```
812 }%
813 }
```

`\ZREF@@extract`

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

`\ZREF@wu@extract`

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

`\zref@extract`

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

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

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

`\ZREF@wu@extractdefault`

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

`\zref@extractdefault`

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

`\zref@def@extract`

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

`\ZREF@def@extract`

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

`\zref@def@extractdefault`

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

`\ZREF@def@extractdefault`

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

`\ZREF@wrapper@unexpanded`

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

`\zref@wrapper@unexpanded`

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

6.2.10 Compatibility with babel

`\zref@wrapper@babel`

```
882 \ZREF@Robust{\long\def}\zref@wrapper@babel#1#2{%  
883 \ifcsname if@safe@actives\endcsname  
884 \expandafter\ltx@firstofone  
885 \else  
886 \expandafter\ltx@secondoftwo  
887 \fi  
888 {%  
889 \if@safe@actives  
890 \expandafter\ltx@secondoftwo  
891 \else  
892 \expandafter\ltx@firstoftwo  
893 \fi  
894 {%  
895 \begingroup  
896 \csname @safe@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 appearance of the latest counter that is incremented by `\refstepcounter`

page the appearance 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 [2016/05/21 v2.26 Module user for zref (HO)]%
963 \RequirePackage{zref-base}[2016/05/21]
964 \ifx\ZREF@base@ok Y%
965 \else
966 \expandafter\endinput
967 \fi
```

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

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

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

```

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

`\zkvlabel`

```

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

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

```

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

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

```

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

`\zrefused` For the following expandible user macros `\zrefused` should be used to notify \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 [2016/05/21 v2.26 Module abspage for zref (HO)]%
1001 \RequirePackage{zref-base}[2016/05/21]
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 [2016/05/21 v2.26 Module counter for zref (HO)]%
1022 \RequirePackage{zref-base}[2016/05/21]
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 [2016/05/21 v2.26 Module lastpage for zref (HO)]%
1042 \RequirePackage{zref-base}[2016/05/21]
1043 \RequirePackage{zref-abspage}[2016/05/21]
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 [2016/05/21 v2.26 Module thepage for zref (HO)]%
1085 \RequirePackage{zref-base}[2016/05/21]
1086 \ifx\ZREF@base@ok Y%
1087 \else
1088   \expandafter\endinput
1089 \fi

1090 \RequirePackage{atbegshi}[2011/10/05]
1091 \RequirePackage{zref-abspage}[2016/05/21]

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 [2016/05/21 v2.26 Module nextpage for zref (HO)]%
1125 \RequirePackage{zref-base}[2016/05/21]
1126 \ifx\ZREF@base@ok Y%
1127 \else
1128 \expandafter\endinput
1129 \fi

1130 \RequirePackage{zref-abspage}[2016/05/21]
1131 \RequirePackage{zref-thepage}[2016/05/21]
1132 \RequirePackage{zref-lastpage}[2016/05/21]
1133 \RequirePackage{uniquecounter}[2009/12/18]

1134 \UniqueCounterNew{znextpage}
1135
1136 \newcommand*{\znextpagesetup}{%
1137 \afterassignment\ZREF@np@setup@i
1138 \def\ZREF@np@call@unknown###1%
1139 }
1140 \def\ZREF@np@setup@i{%
1141 \afterassignment\ZREF@np@setup@ii
1142 \def\ZREF@np@call@nonext###1%
1143 }
1144 \def\ZREF@np@setup@ii{%
1145 \def\ZREF@np@call@next###1%
1146 }
1147 \def\ZREF@np@call@unknown#1{#1}

```

```

1148 \def\ZREF@np@call@nonext#1{#1}
1149 \def\ZREF@np@call@next#1{#1}
1150 \ZREF@ifDefinable\znextpage\def{%
1151   {\UniqueCounterCall{znextpage}{\ZREF@nextpage}}}%
1152 }%
1153 \newcommand*{\znonextpagename}{ }
1154 \newcommand*{\zunknownnextpagename}{\Z@D@page}
1155 \def\ZREF@nextpage#1{%
1156   \begingroup
1157   \def\ZREF@refname@this{zref@np#1}%
1158   \zref@labelbyprops\ZREF@refname@this{abspage}%
1159   \chardef\ZREF@call=0 % unknown
1160   \ZREF@ifrefundefined\ZREF@refname@this{%
1161     }{%
1162     \edef\ZREF@pagenum@this{%
1163       \zref@extractdefault\ZREF@refname@this{abspage}{0}%
1164     }%
1165     \edef\ZREF@refname@next{%
1166       \zref@thepage@name{%
1167         \the\numexpr\ZREF@pagenum@this+1%
1168       }%
1169     }%
1170     \ifnum\ZREF@pagenum@this>0 %
1171       \ZREF@ifrefundefined{LastPage}{%
1172         \zref@ifrefundefined\ZREF@refname@next{%
1173           }{%
1174             \chardef\ZREF@call=2 % next page
1175           }%
1176         }{%
1177           \edef\ZREF@pagenum@last{%
1178             \zref@extractdefault{LastPage}{abspage}{0}%
1179           }%
1180           \ifnum\ZREF@pagenum@this<\ZREF@pagenum@last\ltx@space
1181             \ZREF@ifrefundefined\ZREF@refname@next{%
1182               }{%
1183                 \chardef\ZREF@call=2 % next page
1184               }%
1185             \else
1186               \ifnum\ZREF@pagenum@this=\ZREF@pagenum@this\ltx@space
1187                 \chardef\ZREF@call=1 % no next page
1188               \fi
1189             \fi
1190           }%
1191         \fi
1192       }%
1193     \edef\x{%
1194       \endgroup
1195       \ifcase\ZREF@call
1196         \noexpand\ZREF@np@call@unknown{%
1197           \noexpand\zunknownnextpagename
1198         }%
1199       \or
1200         \noexpand\ZREF@np@call@nonext{%
1201           \noexpand\znonextpagename
1202         }%
1203       \else
1204         \noexpand\ZREF@np@call@next{%
1205           \noexpand\zref@extract{\ZREF@refname@next}{page}%
1206         }%
1207       \fi
1208     }%
1209   \x

```

```

1210 }
1211 </nextpage>

```

6.9 Module `totpages`

```

1212 <*totpages>
1213 \NeedsTeXFormat{LaTeX2e}
1214 \ProvidesPackage{zref-totpages}%
1215 [2016/05/21 v2.26 Module totpages for zref (HO)]%
1216 \RequirePackage{zref-base}[2016/05/21]
1217 \ifx\ZREF@base@ok Y%
1218 \else
1219 \expandafter\endinput
1220 \fi

```

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

```

1221 \RequirePackage{zref-abspage}[2016/05/21]
1222 \RequirePackage{zref-lastpage}[2016/05/21]

```

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

```

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

```

Also we mark the reference `LastPage` as used:

```

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

```

6.10 Module `pagelayout`

```

1230 <*pagelayout>
1231 \NeedsTeXFormat{LaTeX2e}
1232 \ProvidesPackage{zref-pagelayout}%
1233 [2016/05/21 v2.26 Module pagelayout for zref (HO)]%
1234 \RequirePackage{zref-base}[2016/05/21]
1235 \ifx\ZREF@base@ok Y%
1236 \else
1237 \expandafter\endinput
1238 \fi

1239 \RequirePackage{zref-thepage}[2016/05/21]
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}[0pt]{\AtBeginShipoutBoxHeight}

```

```
1318 \zref@newprop*{outputboxdp}[0pt]{\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\csname##1@##2\endcsname{##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 [2016/05/21 v2.26 Module pageattr for zref (HO)]%
1418 \RequirePackage{zref-base}[2016/05/21]
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}[2016/05/21]%
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}[2016/05/21]%
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}[2016/05/21]
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   \fi
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@ifundefined{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 [2016/05/21 v2.26 Module marks for zref (HO)]%
1581 \RequirePackage{zref-base}[2016/05/21]
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 [2016/05/21 v2.26 Module runs for zref (HO)]%

```

\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@filesw
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 [2016/05/21 v2.26 Module perpage for zref (HO)]%
1692 \RequirePackage{zref-base}[2016/05/21]
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}[2016/05/21]
```

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@zerl}}%
```

```
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 [2016/05/21 v2.26 Module titleref for zref (HO)]%
1775 \RequirePackage{zref-base}[2016/05/21]
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.

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

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 removed. 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 accidentally.

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 `\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 ε -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 }%
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 </titleref>
2015 <*xr>
```

6.16 Module xr

```

2016 \NeedsTeXFormat{LaTeX2e}
2017 \ProvidesPackage{zref-xr}%
2018 [2016/05/21 v2.26 Module xr for zref (HO)]%
2019 \RequirePackage{zref-base}[2016/05/21]
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 \tex@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\tx@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\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

```



```

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 [2016/05/21 v2.26 Module hyperref for zref (HO)]%
2422 \RequirePackage{zref-base}[2016/05/21]
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 [2016/05/21 v2.26 Module savepos for zref (HO)]%
2434 \RequirePackage{zref-base}[2016/05/21]
2435 \ifx\ZREF@base@ok Y%
2436 \else
2437 \expandafter\endinput
2438 \fi

LuaTeX compatibility
2439 \ifx\pdfsavepos\@undefined
2440 \let\pdfsavepos \savepos
2441 \let\pdfastxpos \lastxpos
2442 \let\pdfastypos \lastypos
2443 \fi

```

6.18.2 Availability

First we check, whether the feature is available.

```
2444 \ltx@ifundefined{pdfsavepos}{%
2445   \PackageError\ZREF@name{%
2446     \string\pdfsavepos\space is not supported.\MessageBreak
2447     It is provided by pdfTeX (1.40) or XeTeX%
2448   }\ZREF@UpdatePdfTeX
2449   \endinput
2450 }{}%
```

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 `XƎTeX` also provides `\pdfsavepos`.

```
2451 \RequirePackage{ifpdf}
2452 \ifpdf
2453 \else
2454   \ltx@ifundefined{pdftexversion}{%
2455     }{%
2456     \ifnum\pdftexversion<140 %
2457       \PackageError\ZREF@name{%
2458         \string\pdfsavepos\space is not supported in DVI mode%
2459         \MessageBreak
2460         of this pdfTeX version%
2461       }\ZREF@UpdatePdfTeX
2462       \expandafter\expandafter\expandafter\endinput
2463     \fi
2464   }%
2465 \fi
```

6.18.3 Setup

```
2466 \zref@newlist{savepos}
2467 \zref@newprop*{posx}{0}{\the\pdfastxpos}
2468 \zref@newprop*{posy}{0}{\the\pdfastypos}
2469 \zref@addprops{savepos}{posx,posy}
```

6.18.4 User macros

`\zref@savepos`

```
2470 \def\zref@savepos{%
2471   \if@filesw
2472     \pdfsavepos
2473   \fi
2474 }
```

`\ZREF@zsavepos`

```
2475 \def\ZREF@zsavepos#1#2#3{%
2476   \@bsphack
2477   \if@filesw
2478     \zref@savepos
2479     #1{#3}{#2}%
2480   \ltx@ifundefined{TeXXeTstate}{%
2481     }{%
2482     \ifnum\TeXXeTstate=\ltx@zero
2483       \else
2484         \zref@savepos
2485       \fi
2486     }%
2487   \fi
2488   \@esphack
2489 }
```

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

```
2490 \ZREF@ifDefinable\zsavapos\def{%
2491   {%
2492     \ZREF@zsavapos\zref@labelbylist{savepos}%
2493   }%
2494 }
```

`\zsaveposx`

```
2495 \ZREF@ifDefinable\zsaveposx\def{%
2496   {%
2497     \ZREF@zsavepos\zref@labelbyprops{posx}%
2498   }%
2499 }
```

`\zsaveposy`

```
2500 \ZREF@ifDefinable\zsaveposy\def{%
2501   {%
2502     \ZREF@zsavepos\zref@labelbyprops{posy}%
2503   }%
2504 }
```

`\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 pdf \TeX . Use
differences instead. The unit of the position numbers is sp.

```
2505 \newcommand*{\zposx}[1]{%
2506   \zref@extract{#1}{posx}%
2507 }%
2508 \newcommand*{\zposy}[1]{%
2509   \zref@extract{#1}{posy}%
2510 }%
```

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 \LaTeX about undefined references.

`\ZREF@savepos@ok`

```
2511 \let\ZREF@savepos@ok=Y
2512 \</savepos>
```

6.19 Module `abspos`

6.19.1 Identification

```
2513 <*abspos>
2514 \NeedsTeXFormat{LaTeX2e}
2515 \ProvidesPackage{zref-abspos}%
2516 [2016/05/21 v2.26 Module abspos for zref (HO)]%
2517 \RequirePackage{zref-base}[2016/05/21]
2518 \ifx\ZREF@base@ok Y%
2519 \else
2520 \expandafter\endinput
2521 \fi
2522 \RequirePackage{zref-savepos}[2016/05/21]
2523 \ifx\ZREF@savepos@ok Y%
2524 \else
2525 \expandafter\endinput
2526 \fi
2527 \RequirePackage{zref-pagelayout}[2016/05/21]
```

```

2528 \zref@addprop{savepos}{abspage}
2529 \RequirePackage{ifpdf}[2010/09/13]

\zref@absposx
2530 \newcommand*{\zref@absposx}[3]{%
2531   \number
2532   \expandafter\zref@absposnumx\expandafter{%
2533     \number\zref@extractdefault{#1}{abspage}{0}%
2534     }{#2}{#3}%
2535   \ltx@space
2536 }

\zref@absposy
2537 \newcommand*{\zref@absposy}[3]{%
2538   \number
2539   \expandafter\zref@absposnumy\expandafter{%
2540     \number\zref@extractdefault{#1}{abspage}{0}%
2541     }{#2}{#3}%
2542   \ltx@space
2543 }

\zref@absposnumx
2544 \newcommand*{\zref@absposnumx}[3]{%
2545   \number
2546   % \ifnum#1>\ltx@zero
2547   % \zref@ifundefined{thepage#1}{%
2548   %   0%
2549   % }{%
2550   % \numexpr\ZREF@absposnum{thepage#1}{#2}{x}{#3}\relax
2551   % }%
2552   % \else
2553   %   0%
2554   % \fi
2555 }

\zref@absposnumy
2556 \newcommand*{\zref@absposnumy}[3]{%
2557   \number
2558   % \ifnum#1>\ltx@zero
2559   % \zref@ifundefined{thepage#1}{%
2560   %   0%
2561   % }{%
2562   % \numexpr\ZREF@absposnum{thepage#1}{#2}{y}{#3}\relax
2563   % }%
2564   % \else
2565   %   0%
2566   % \fi
2567 }

\ZREF@absposnum
2568 \def\ZREF@absposnum#1#2#3#4{%
2569   \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
2570     0%
2571   }{%
2572     \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
2573   }%
2574 }

\zref@def@absposx
2575 \ZREF@Robust\def\zref@def@absposx#1{%
2576   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposx}%
2577 }

```

```

\zref@def@absposy
2578 \ZREF@Robust\def\zref@def@absposy#1{%
2579 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
2580 }

\zref@def@absposnumx
2581 \ZREF@Robust\def\zref@def@absposnumx#1{%
2582 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumx}%
2583 }

\zref@def@absposnumy
2584 \ZREF@Robust\def\zref@def@absposnumy#1{%
2585 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumy}%
2586 }

\ZREF@def@abspos
2587 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2588 \edef#1{#2{#3}{#4}{#5}}%
2589 }

\zref@absposused
2590 \ZREF@Robust\def\zref@absposused{%
2591 \zref@wrapper@babel\ZREF@abspos@used
2592 }

\ZREF@abspos@used
2593 \def\ZREF@abspos@used#1{%
2594 \zref@refused{#1}%
2595 \zref@ifrefundefined{#1}{%
2596 }{%
2597 \begingroup
2598 \edef\ZREF@temp{%
2599 \zref@extractdefault{#1}{abspage}{0}%
2600 }%
2601 \ifnum\ZREF@temp>\tx@zero
2602 \zref@refused{thepage\ZREF@temp}%
2603 \else
2604 \@PackageError{zref-abspos}{%
2605 \string\zref@pos@label@used\tx@space
2606 needs property `abspage'\MessageBreak
2607 in label `#1'%
2608 }\@ehc
2609 \fi
2610 \endgroup
2611 }%
2612 }

\zref@absposnumused
2613 \newcommand*\zref@absposnumused}[1]{%
2614 \ifnum#1>\tx@zero
2615 \zref@refused{thepage\number#1}%
2616 \else
2617 \@PackageError{zref-abspos}{%
2618 Invalid absolute page number (#1)\MessageBreak
2619 for \string\zref@pos@num@used.\MessageBreak
2620 A positive integer is expected%
2621 }\@ehc
2622 \fi
2623 }

```

`\zref@ifabsposundefined`

```
2624 \def\zref@ifabsposundefined#1{%
2625   \zref@ifrefundefined{#1}\ltx@firsttwo{%
2626     \expandafter\zref@ifabsposnumundefined\expandafter{%
2627       \number\zref@extractdefault{#1}{abspage}{0}%
2628     }%
2629   }%
2630 }
```

`\zref@ifabsposnumundefined`

```
2631 \def\zref@ifabsposnumundefined#1{%
2632   \ifnum\ZREF@number{#1}>\ltx@zero
2633     \zref@ifrefundefined{thepage#1}%
2634     \ltx@firstoftwo\ltx@secondoftwo
2635   \else
2636     \expandafter\ltx@firstoftwo
2637   \fi
2638 }
```

6.19.2 Media

`\ZREF@abspos@media@width`

```
2639 \edef\ZREF@abspos@media@width{%
2640   \ltx@ifundefined{pdfpagewidth}{%
2641     \ltx@ifundefined{mediawidth}{%
2642       \ltx@ifundefined{stockwidth}{%
2643         paperwidth%
2644       }{%
2645         stockwidth%
2646       }%
2647     }{%
2648       mediawidth%
2649     }%
2650   }{%
2651     pdfpagewidth%
2652   }%
2653 }
```

`\ZREF@abspos@media@height`

```
2654 \edef\ZREF@abspos@media@height{%
2655   \ltx@ifundefined{pdfpageheight}{%
2656     \ltx@ifundefined{mediaheight}{%
2657       \ltx@ifundefined{stockheight}{%
2658         paperheight%
2659       }{%
2660         stockheight%
2661       }%
2662     }{%
2663       mediaheight%
2664     }%
2665   }{%
2666     \noexpand\ifcase\pdfpageheight
2667       \ltx@ifundefined{stockheight}{%
2668         paperheight%
2669       }{%
2670         stockheight%
2671       }%
2672     \noexpand\else
2673       pdfpageheight%
2674     \noexpand\fi
2675   }%
2676 }
```

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

```
2677 \def\ZREF@abspos@media@x@left#1{%  
2678 0%  
2679 }
```

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

```
2680 \def\ZREF@abspos@media@x@right#1{%  
2681 \zref@extract{#1}\ZREF@abspos@media@width  
2682 }
```

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

```
2683 \def\ZREF@abspos@media@x@center#1{%  
2684 \ZREF@abspos@media@x@left{#1}%  
2685 +\zref@extract{#1}\ZREF@abspos@media@width/2%  
2686 }
```

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

```
2687 \def\ZREF@abspos@media@y@top#1{%  
2688 \zref@extract{#1}\ZREF@abspos@media@height  
2689 }
```

`ZREF@abspos@media@y@bottom`

```
2690 \def\ZREF@abspos@media@y@bottom#1{%  
2691 0%  
2692 }
```

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

```
2693 \def\ZREF@abspos@media@y@center#1{%  
2694 \zref@extract{#1}\ZREF@abspos@media@height/2%  
2695 }
```

6.19.3 Paper

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

```
2696 \def\ZREF@abspos@paper@x@left#1{%  
2697 0%  
2698 }
```

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

```
2699 \def\ZREF@abspos@paper@x@right#1{%  
2700 \zref@extract{#1}{paperwidth}%  
2701 }
```

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

```
2702 \def\ZREF@abspos@paper@x@center#1{%  
2703 \zref@extract{#1}{paperwidth}/2%  
2704 }
```

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

```
2705 \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top
```

`ZREF@abspos@paper@y@bottom`

```
2706 \def\ZREF@abspos@paper@y@bottom#1{%  
2707 \ZREF@abspos@paper@y@top{#1}%  
2708 -\zref@extract{#1}{paperheight}%  
2709 }
```

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

```
2710 \def\ZREF@abspos@paper@y@center#1{%  
2711 \ZREF@abspos@paper@y@top{#1}%  
2712 -\zref@extract{#1}{paperheight}/2%  
2713 }
```

6.19.4 Origin

`\ZREF@abspos@origin@x`

```
2714 \let\ZREF@temp\ltx@two
2715 \ltx@ifundefined{pdfhorigin}{}{%
2716   \ifpdf
2717     \let\ZREF@temp\ltx@zero
2718   \fi
2719 }
2720 \ifx\ZREF@temp\ltx@two
2721   \ifnum\mag=1000 %
2722     \let\ZREF@temp\ltx@one
2723   \fi
2724 \fi
2725 \ifcase\ZREF@temp
2726   \def\ZREF@abspos@origin@x#1{%
2727     \zref@extract{#1}{pdfhorigin}%
2728   }%
2729 \or
2730   \def\ZREF@abspos@origin@x#1{%
2731     4736286%
2732   }%
2733 \or
2734   \def\ZREF@abspos@origin@x#1{%
2735     \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2736   }%
2737 \fi
```

`\ZREF@abspos@origin@y`

```
2738 \let\ZREF@temp\ltx@two
2739 \ltx@ifundefined{pdfvorigin}{}{%
2740   \ifpdf
2741     \let\ZREF@temp\ltx@zero
2742   \fi
2743 }
2744 \ifx\ZREF@temp\ltx@two
2745   \ifnum\mag=1000 %
2746     \let\ZREF@temp\ltx@one
2747   \fi
2748 \fi
2749 \ifcase\ZREF@temp
2750   \def\ZREF@abspos@origin@y#1{%
2751     \zref@extract{#1}{pdfvorigin}%
2752   }%
2753 \or
2754   \def\ZREF@abspos@origin@y#1{%
2755     4736286%
2756   }%
2757 \or
2758   \def\ZREF@abspos@origin@y#1{%
2759     \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2760   }%
2761 \fi
```

6.19.5 Header

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

```
2762 \def\ZREF@abspos@head@x@left#1{%
2763   \ZREF@abspos@paper@x@left{#1}%
2764   +\ZREF@abspos@origin@x{#1}%
2765   +\zref@extract{#1}{hoffset}%
2766   +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
```

```

2767 \zref@extract{#1}{oddsidemargin}%
2768 \else
2769 \zref@extract{#1}{evensidemargin}%
2770 \fi
2771 }

```

`\ZREF@abspos@head@x@right`

```

2772 \def\ZREF@abspos@head@x@right#1{%
2773 \ZREF@abspos@head@x@left{#1}%
2774 +\zref@extract{#1}{textwidth}%
2775 }

```

`\ZREF@abspos@head@x@center`

```

2776 \def\ZREF@abspos@head@x@center#1{%
2777 \ZREF@abspos@head@x@left{#1}%
2778 +\zref@extract{#1}{textwidth}/2%
2779 }

```

`\ZREF@abspos@head@y@top`

```

2780 \def\ZREF@abspos@head@y@top#1{%
2781 \ZREF@abspos@paper@y@top{#1}%
2782 -\ZREF@abspos@origin@y{#1}%
2783 -\zref@extract{#1}{voffset}%
2784 -\zref@extract{#1}{topmargin}%
2785 }

```

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

```

2786 \def\ZREF@abspos@head@y@bottom#1{%
2787 \ZREF@abspos@head@y@top{#1}%
2788 -\zref@extract{#1}{headheight}%
2789 }

```

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

```

2790 \def\ZREF@abspos@head@y@center#1{%
2791 \ZREF@abspos@head@y@top{#1}%
2792 -\zref@extract{#1}{headheight}/2%
2793 }

```

6.19.6 Body

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

```

2794 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left

```

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

```

2795 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right

```

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

```

2796 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center

```

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

```

2797 \def\ZREF@abspos@body@y@top#1{%
2798 \ZREF@abspos@head@y@bottom{#1}%
2799 -\zref@extract{#1}{headsep}%
2800 }

```

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

```

2801 \def\ZREF@abspos@body@y@bottom#1{%
2802 \ZREF@abspos@body@y@top{#1}%
2803 -\zref@extract{#1}{textheight}%
2804 }

```

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

```
2805 \def\ZREF@abspos@body@y@center#1{%
2806 \ZREF@abspos@body@y@top{#1}%
2807 -\zref@extract{#1}{textheight}/2%
2808 }
```

6.19.7 Footer

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

```
2809 \let\ZREF@abspos@foot@x@left\ZREF@abspos@head@x@left
```

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

```
2810 \let\ZREF@abspos@foot@x@right\ZREF@abspos@head@x@right
```

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

```
2811 \let\ZREF@abspos@foot@x@center\ZREF@abspos@head@x@center
```

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

```
2812 \def\ZREF@abspos@foot@y@bottom#1{%
2813 \ZREF@abspos@body@y@bottom{#1}%
2814 -\zref@extract{#1}{footskip}%
2815 }
```

6.19.8 Marginal notes

`ZREF@abspos@marginpar@x@left`

```
2816 \def\ZREF@abspos@marginpar@x@left#1{%
2817 \ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2818 \ZREF@abspos@body@x@right{#1}%
2819 +\zref@extract{#1}{marginparsep}%
2820 \else
2821 \ZREF@abspos@body@x@left{#1}%
2822 -\zref@extract{#1}{marginparsep}%
2823 -\zref@extract{#1}{marginparwidth}%
2824 \fi
2825 }
```

`ZREF@abspos@marginpar@x@right`

```
2826 \def\ZREF@abspos@marginpar@x@right#1{%
2827 \ZREF@abspos@marginpar@x@left{#1}%
2828 +\zref@extract{#1}{marginparwidth}%
2829 }
```

`ZREF@abspos@marginpar@x@center`

```
2830 \def\ZREF@abspos@marginpar@x@center#1{%
2831 \ZREF@abspos@marginpar@x@left{#1}%
2832 +\zref@extract{#1}{marginparwidth}/2%
2833 }
```

`ZREF@abspos@marginpar@y@top`

```
2834 \let\ZREF@abspos@marginpar@y@top\ZREF@abspos@body@y@top
```

`ZREF@abspos@marginpar@y@bottom`

```
2835 \let\ZREF@abspos@marginpar@y@bottom\ZREF@abspos@body@y@bottom
```

`ZREF@abspos@marginpar@y@center`

```
2836 \let\ZREF@abspos@marginpar@y@center\ZREF@abspos@body@y@center
```

6.19.9 Stock paper

```
\ZREF@abspos@stock@x@left
2837 \let\ZREF@abspos@stock@x@left\ZREF@abspos@paper@x@left

\ZREF@abspos@stock@x@right
2838 \let\ZREF@abspos@stock@x@right\ZREF@abspos@paper@x@right

\ZREF@abspos@stock@x@center
2839 \let\ZREF@abspos@stock@x@center\ZREF@abspos@paper@x@center

\ZREF@abspos@stock@y@top
2840 \let\ZREF@abspos@stock@y@top\ZREF@abspos@paper@y@top

\ZREF@abspos@stock@y@bottom
2841 \let\ZREF@abspos@stock@y@bottom\ZREF@abspos@paper@y@bottom

\ZREF@abspos@stock@y@center
2842 \let\ZREF@abspos@stock@y@center\ZREF@abspos@paper@y@center

2843 </abspos>
```

6.20 Module dotfill

```
2844 <*dotfill>
2845 \NeedsTeXFormat{LaTeX2e}
2846 \ProvidesPackage{zref-dotfill}%
2847 [2016/05/21 v2.26 Module dotfill for zref (HO)]%
2848 \RequirePackage{zref-base}[2016/05/21]
2849 \ifx\ZREF@base@ok Y%
2850 \else
2851 \expandafter\endinput
2852 \fi
```

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

```
2853 \RequirePackage{zref-savepos}[2016/05/21]
```

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

```
2854 \zref@require@unique
```

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

```
2855 \RequirePackage{keyval}
```

The definitions of the keys follow.

```
2856 \define@key{ZREF@DF}{unit}{%
2857 \def\ZREF@df@unit{#1}%
2858 }
2859 \define@key{ZREF@DF}{min}{%
2860 \def\ZREF@df@min{#1}%
2861 }
2862 \define@key{ZREF@DF}{dot}{%
2863 \def\ZREF@df@dot{#1}%
2864 }
```

Defaults are set, see user interface.

```
2865 \providecommand\ZREF@df@min{2}
2866 \providecommand\ZREF@df@unit{.44em}
2867 \providecommand\ZREF@df@dot{.}
```

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

```
2868 \newcommand*{\zdotfillsetup}{\kvsetkeys{ZREF@DF}}
```

`\zdotfill` `\zdotfill` sets labels at the left and the right to get the horizontal position. `\zsavepos` is not used, because we do not need the vertical position.

```

2869 \ZREF@ifDefinable\zdotfill\def{%
2870  {%
2871   \leavevmode
2872   \global\advance\c@zref@unique\ltx@one
2873   \begingroup
2874   \def\ZREF@temp{zref@\number\c@zref@unique}%
2875   \pdfsavepos
2876   \zref@labelbyprops{\thezref@unique L}{posx}%
2877   \setlength{\dimen@}{\ZREF@df@unit}%
2878   \zref@ifrefundefined{\thezref@unique R}{%
2879     \ZREF@dotfill
2880   }{%
2881     \ifnum\numexpr\zposx{\thezref@unique R}%
2882       -\zposx{\thezref@unique L}\relax
2883       <\dimexpr\ZREF@df@min\dimen@\relax
2884       \hfill
2885     \else
2886       \ZREF@dotfill
2887     \fi
2888   }%
2889   \pdfsavepos
2890   \zref@labelbyprops{\thezref@unique R}{posx}%
2891 \endgroup
2892 \kern\z@
2893 }%
2894 }

```

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

```

2895 \def\ZREF@dotfill{%
2896 \cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill
2897 }
2898 </dotfill>

```

6.21 Module `env`

```

2899 <*env>
2900 \NeedsTeXFormat{LaTeX2e}
2901 \ProvidesPackage{zref-env}%
2902 [2016/05/21 v2.26 Module env for zref (HO)]%
2903 \RequirePackage{zref-base}[2016/05/21]
2904 \ifx\ZREF@base@ok Y%
2905 \else
2906 \expandafter\endinput
2907 \fi
2908 \zref@newprop{envname}[]{\@currenvir}
2909 \zref@newprop{envline}[]{\zref@env@line}

```

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

```

2910 \def\zref@env@line{%
2911 \ifx\@currenvline\ltx@empty
2912 \else
2913 \expandafter
2914 \ZREF@ENV@line\@currenvline\ltx@empty line \ltx@empty\@nil
2915 \fi
2916 }

```

`\ZREF@ENV@line`

```

2917 \def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%
2918 </env>

```

7 Test

7.1 \zref@localaddprop

```
2919 ⟨*test1⟩
2920 \NeedsTeXFormat{LaTeX2e}
2921 \nofiles
2922 \documentclass{article}
2923 \usepackage{zref-base}[2016/05/21]
2924 \usepackage{qstest}
2925 \IncludeTests{*}
2926 \LogTests{log}{*}{*}
2927
2928 \makeatletter
2929 \def\ExpectList#1#2{%
2930   \expandafter\expandafter\expandafter\Expect
2931   \expandafter\expandafter\expandafter{\csname Z@L@#1\endcsname}{#2}%
2932 }
2933 \begin{qstest}{localaddprop}{localaddprop}
2934   \ExpectList{main}{\default\page}%
2935   \Expect{undefined}*\{\meaning\foobar}%
2936   \zref@newprop{foobar}{FOO}%
2937   \Expect{undefined}*\{\meaning\foobar}%
2938   \zref@newlist{alist}%
2939   \ExpectList{alist}{}%
2940   \begingroup
2941     \zref@localaddprop{main}{foobar}%
2942     \Expect{undefined}*\{\meaning\foobar}%
2943     \ExpectList{main}{\default\page\foobar}%
2944     \zref@localaddprop{alist}{page}%
2945     \ExpectList{alist}{\page}%
2946   \endgroup
2947   \ExpectList{main}{\default\page}%
2948   \ExpectList{alist}{}%
2949   \zref@addprop{alist}{foobar}%
2950   \ExpectList{alist}{\foobar}%
2951   \Expect{undefined}*\{\meaning\foobar}%
2952 \end{qstest}
2953 \@@end
2954 ⟨/test1⟩
```

7.2 Module base

```
2955 ⟨*test-base⟩
2956 \NeedsTeXFormat{LaTeX2e}
2957 \documentclass{article}
2958 \usepackage{zref-base,zref-titleref}[2016/05/21]
2959 \usepackage{qstest}
2960 \IncludeTests{*}
2961 \LogTests{log}{*}{*}
2962
2963 \makeatletter
2964 \newcommand*\DefExpand}[2]{%
2965   \expandafter\expandafter\expandafter\def
2966   \expandafter\expandafter\expandafter#1%
2967   \expandafter\expandafter\expandafter{#2}%
2968   \@onelevel@sanitize#1%
2969 }
2970 \newcommand*\Test}[3]{%
2971   \Expect{#2}*\{#1}%
2972   \zref@wrapper@unexpanded{%
2973     \Expect*{#3}*\{#1}%
2974   }%
2975   \DefExpand\x{#1}%

```

```

2976 \Expect*{#3}*{\x}%
2977 }
2978 \makeatother
2979
2980 \begin{document}
2981 \section{\textit{Hello} \textbf{World}}
2982 \label{sec:hello}
2983 \makeatletter
2984 \zref@newprop{foo}[\@empty D\@empty efault]{\@empty V\@empty alue}
2985 \begin{qstest}{getcurrent}{getcurrent}
2986 \Test{\zref@getcurrent{foo}}%
2987     {Value}{\noexpand\@empty V\noexpand\@empty alue}%
2988 \Test{\zref@getcurrent{xy}}{}{}%
2989 \end{qstest}
2990 \begin{qstest}{extract}{extract}
2991 \def\textbf#1{<#1>}%
2992 \def\textit#1{[#1]}% hash-ok
2993 \Test{\zref@extractdefault{xy}{page}{\@empty D\@empty efault}}%
2994     {Default}{\noexpand\@empty D\noexpand\@empty efault}%
2995 \Test{\zref@extractdefault{sec:hello}{foo}{\@empty A\@empty B}}%
2996     {AB}{\noexpand\@empty A\noexpand\@empty B}%
2997 \Test{\zref@extract{sec:hello}{foo}}%
2998     {Default}{\noexpand\@empty D\noexpand\@empty efault}%
2999 \zref@ifrefundefined{sec:hello}{%
3000 }{%
3001 \Test{\zref@extract{sec:hello}{default}}{1}{1}%
3002 \Test{\zref@extract{sec:hello}{title}}%
3003     {[Hello] <World>}%
3004     {\noexpand\textit{Hello} \noexpand\textbf{World}}%
3005 }%
3006 \end{qstest}
3007 \end{document}
3008 </test-base>

```

7.3 Module runs

```

3009 <*test-runs>
3010 \NeedsTeXFormat{LaTeX2e}
3011 \documentclass{article}
3012 \usepackage{zref-runs}[2016/05/21]
3013 \usepackage{qstest}
3014 \IncludeTests{*}
3015 \LogTests{log}{*}{*}
3016
3017 \begin{qstest}{zruns-preamble}{zruns-preamble}
3018 \Expect{0}*{\zruns}%
3019 \end{qstest}
3020
3021 \AtBeginDocument{%
3022 \begin{qstest}{zruns-atbeginndocument}{zruns-atbeginndocument}%
3023 \Expect*{\number\ExpectRuns}*{\zruns}%
3024 \end{qstest}%
3025 }
3026
3027 \begin{document}
3028 \begin{qstest}{zruns-document}{zruns-document}
3029 \Expect*{\number\ExpectRuns}*{\zruns}%
3030 \end{qstest}
3031 \end{document}
3032 </test-runs>

```

7.4 Module titleref

```

3033 <*test-titleref-memoir>

```

```

3034 \NeedsTeXFormat{LaTeX2e}
3035 \documentclass{memoir}
3036 \usepackage{zref-titleref}[2016/05/21]
3037 \usepackage{qstest}
3038 \IncludeTests{*}
3039 \LogTests{log}{*}{*}
3040 \begin{document}
3041 \makeatletter
3042 \def\List{}
3043 \def\Label#1{%
3044   \zref@label{#1}%
3045   \g@addto@macro\List{%
3046     \par
3047     #1: [\ztitleref{#1}]%
3048   }%
3049   \mbox{}}%
3050 \zref@refused{#1}%
3051 \zref@ifrefundefined{#1}{%
3052 }{%
3053   \begingroup
3054   \edef\x{\zref@extract{#1}{title}}%
3055   \Expect{OK/}*{\expandafter\ltx@carthree\x{}{}{}\@nil}%
3056   \endgroup
3057 }%
3058 }
3059 \def\Test#1{%
3060   \csname#1\endcsname*{OK/#1}%
3061   \Label{#1*}%
3062   \csname#1\endcsname{OK/#1}%
3063   \Label{#1}%
3064   \csname#1\endcsname[OK/#1-toc]%
3065     {WRONG-in-titleref/#1-toc-2}%
3066   \Label{#1-toc}%
3067   \expandafter\ifx\csname#1\endcsname\part
3068   \else
3069     \headnamereffalse
3070     \csname#1\endcsname[OK/#1-th-toc]%
3071       [WRONG-in-titleref/#1-th-toc-2]%
3072       {WRONG-in-titleref/#1-th-toc-3}%
3073     \Label{#1-th-toc}%
3074     \headnamereftrue
3075     \csname#1\endcsname[WRONG-in-titleref/#1-th-head-1]%
3076       [OK/#1-th-head]%
3077       {WRONG-in-titleref/#1-th-head-3}%
3078     \Label{#1-th-head}%
3079   \fi
3080 }
3081 \begin{qstest}{section}{section}
3082 \@for\x:=part,chapter,section,subsection,subsubsection\do{%
3083   \expandafter\Test\expandafter{\x}%
3084 }%
3085 \end{qstest}
3086 \newpage
3087 \List
3088 \end{document}
3089 </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/
```

8.3 Package installation

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

```
tex zref.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

²<http://ctan.org/pkg/zref>

<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 (`teTEX`, `mikTEX`, ...) relies on file name databases, you must refresh these. For example, `teTEX` users run `texhash` or `mktextlsr`.

8.5 Some details for the interested

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

plain `TEX`: Run `docstrip` and extract the files.

`LATEX`: Generate the documentation.

If you insist on using `LATEX` for `docstrip` (really, `docstrip` does not need `LATEX`), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{zref.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdfLATEX`:

```
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`.

```
3090 <*catalogue>
3091 <?xml version='1.0' encoding='us-ascii'?>
3092 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
3093 <entry datestamp='$Date$' modifier='$Author$' id='zref'>
3094   <name>zref</name>
3095   <caption>A new reference scheme for LaTeX.</caption>
3096   <authorref id='auth:oberdiek' />
3097   <copyright owner='Heiko Oberdiek' year='2006-2012' />
3098   <license type='lppl1.3' />
3099   <version number='2.26' />
3100   <description>
3101     This package offers a means to remove the limitation, of only two
3102     properties, that is inherent in the way LaTeX's reference system
3103     works. The package implements an extensible referencing system,
3104     where properties are handled in a more flexible way. It provides
3105     an interface for macro programmers to access the new reference
3106     scheme and some applications that use it.
3107     <p />
3108     The package is part of the <xref refid='oberdiek'>oberdiek</xref>
3109     bundle.
3110   </description>
3111   <documentation details='Package documentation'
3112     href='ctan:/macros/latex/contrib/oberdiek/zref.pdf' />
3113   <ctan file='true' path='/macros/latex/contrib/oberdiek/zref.dtx' />
3114   <miktex location='oberdiek' />
3115   <texlive location='oberdiek' />
3116   <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip' />
3117 </entry>
3118 </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 Kastrop, 2002/12/20 v1.0. [CTAN:macros/latex/contrib/bigfoot/perpage.dtx](#)
- [6] Package `titleref`, Donald Arsenu, 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 $X_{\text{g}}\text{T}_{\text{E}}\text{X}$.

[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 $\varepsilon\text{-T}_{\text{E}}\text{X}$'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 `toxtlabel` 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.

[2016/05/21 v2.26]

- update zref-savepos for new luatex

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	A
<code>\@@end</code> 2953	<code>\AddLineBeginAux</code> 280
<code>\@PackageError</code> . . . 508, 524, 2604, 2617	<code>\advance</code>
<code>\@PackageInfo</code> 2062	1053, 1393, 1548, 1736, 2058, 2872
<code>\@PackageInfoNoLine</code> 548, 563, 1258, 1346, 1440, 1453, 1465, 1521, 1566	<code>\afterassignment</code> 233, 1137, 1141
<code>\@PackageWarning</code> 691	<code>\AfterLastShipout</code> . . . 1050, 1410, 1573
<code>\@addtoreset</code> 911, 1008	<code>\Alph</code> 7
<code>\@auxout</code> 715	<code>\anchor</code> 2376
<code>\@begintheorem</code> 2001, 2006	<code>\AtBeginDocument</code>
<code>\@bsphack</code> 600, 610, 630, 2476	1029, 1226, 1675, 1844, 3021
<code>\@caption</code> 1846	<code>\AtBeginShipout</code> 1012, 1097
<code>\@chapter</code> 1858, 1895	<code>\AtBeginShipoutAddToBox</code> 1098
<code>\@currentHref</code> 940	<code>\AtBeginShipoutBoxDepth</code> 1318
<code>\@currentlabel</code> 935	<code>\AtBeginShipoutBoxHeight</code> 1317
<code>\@currenvir</code> 2908	<code>\AtBeginShipoutBoxWidth</code> 1316
<code>\@currentline</code> 2911, 2914	<code>\AtEndOfPackage</code> 195
<code>\@ehc</code> 296, 306, 491, 514, 526, 1596, 2608, 2621	<code>\AtVeryEndDocument</code> 1337, 1509
<code>\@empty</code> 2984, 2987, 2993, 2994, 2995, 2996, 2998	B
<code>\@esphack</code> 607, 627, 642, 2488	<code>\beamer@section</code> 1932
<code>\@for</code> 3082	<code>\beamer@subsection</code> 1938
<code>\@ifclassloaded</code> 1893, 1930	<code>\beamer@subsubsection</code> 1944
<code>\@ifdefinable</code> 242, 289	<code>\begin</code> 23, 57, 100, 106, 156, 172, 2933, 2980, 2985, 2990, 3017, 3022, 3027, 3028, 3040, 3081
<code>\@ifnextchar</code> 530, 1712	<code>\bfseries</code> 928
<code>\@ifpackageloaded</code> 1950, 1969, 1977, 1996	C
<code>\@ifstar</code> 495, 2082	<code>\c@abspage</code> 1015, 1744, 1751
<code>\@ifundefined</code> 192, 909, 1723, 2218, 2260	<code>\c@page</code> . . . 954, 1053, 1741, 2766, 2817
<code>\@input</code> 2279	<code>\c@zpage</code> 1741, 1747
<code>\@inputcheck</code> 2110, 2111, 2126, 2163, 2165	<code>\c@zref@unique</code> . . 915, 1736, 2872, 2874
<code>\@latex@warning</code> 777	<code>\ch@pt@c</code> 1899, 1905, 1911
<code>\@mainaux</code> 1681	<code>\chapter</code> 24, 30, 32, 61, 82
<code>\@namedef</code> 535	<code>\ChapterPages</code> 91, 112
<code>\@newl@bel</code> 285	<code>\ChapterStart</code> 78, 135, 150, 166
<code>\@nil</code> 1611, 1613, 1811, 1818, 2000, 2001, 2154, 2157, 2337, 2346, 2914, 2917, 3055	<code>\ChapterStop</code> 85, 148, 165, 184
<code>\@onelevel@sanitize</code>	<code>\chardef</code> 1159, 1174, 1183, 1187
422, 440, 505, 533, 1999, 2002, 2968	<code>\cleaders</code> 2896
<code>\@opargbegintheorem</code> 1991	<code>\cleardoublepage</code> 79, 86
<code>\@part</code> 1852	<code>\clearpage</code> 62
<code>\@schapter</code> 1876	<code>\closein</code> 2126
<code>\@sect</code> 1864	<code>\columnsep</code> 1309
<code>\@spart</code> 1870	<code>\columnwidth</code> 1308
<code>\@ssect</code> 1882	<code>\comma@entry</code> . . 343, 344, 346, 352, 378, 379, 381, 387, 614, 616, 620, 1617, 1618, 1619, 1625, 1628
<code>\@testdef</code> 1328, 1329, 1500, 1501	<code>\comma@parse</code> 342, 377, 613, 1616
<code>\@testopt</code> 2084, 2087, 2097	<code>\count@</code> 1343, 1354, 1355, 1357, 1392, 1393, 1402, 1404, 1405, 1518, 1539, 1541, 1542, 1547, 1548, 2052, 2057, 2068, 2069
<code>\@tfor</code> 322, 720	<code>\csname</code> 252, 253, 290, 315, 316, 317, 326, 351, 352, 369, 370, 386, 387, 404, 405, 425, 427, 444, 462, 465, 478, 536, 538, 539,
<code>\@undefined</code> 1767, 2336, 2439	
<code>\@</code> 25, 26, 27, 28, 153, 155, 157, 158, 170, 173, 2245, 2339, 2364, 2374, 2378, 2394	
<code>_</code> 44, 45	

544, 554, 559, 565, 578, 587, 604, 620, 651, 661, 730, 738, 744, 787, 788, 790, 810, 833, 834, 835, 896, 1333, 1505, 1646, 1654, 1706, 1724, 1726, 1729, 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, 2410, 2572, 2931, 3060, 3062, 3064, 3067, 3070, 3075	\etex@unexpanded .. 591, 819, 839, 2186
\current@chapid	80, 88
D	
\DeclareBoolOption	2037, 2038, 2039, 2043
\DeclareOption	194
\default ...	2357, 2397, 2934, 2943, 2947
\DefExpand	2964, 2975
\define@key	1819, 1822, 1825, 1828, 2040, 2856, 2859, 2862
\descriptionlabel	1888
\detokenize	1806
\dftest	167, 174, 175, 176, 177, 178, 179, 180, 181, 182
\dimen@	2877, 2883, 2896
\dimexpr	153, 155, 1405, 2735, 2759, 2883
\directlua	1250, 1432
\do	327, 720, 3082
\documentclass	4, 39, 68, 272, 2922, 2957, 3011, 3035
\dotfill	169, 173
E	
\emph	150
\end	34, 64, 130, 159, 183, 185, 2952, 2989, 3006, 3007, 3019, 3024, 3030, 3031, 3085, 3088
\endcsname	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, 2572, 2931, 3060, 3062, 3064, 3067, 3070, 3075
\endinput	192, 264, 277, 966, 1004, 1025, 1047, 1088, 1128, 1219, 1237, 1421, 1477, 1584, 1695, 1778, 2022, 2425, 2437, 2449, 2462, 2520, 2525, 2851, 2906
\escapechar	329, 420, 460, 461, 467, 722, 1246, 1273, 1428
\evensidemargin	1300
\Expect	2930, 2935, 2937, 2942, 2951, 2971, 2973, 2976, 3018, 3023, 3029, 3055
\ExpectList	2929, 2934, 2939, 2943, 2945, 2947, 2948, 2950
\ExpectRuns	3023, 3029
\externaldocument	2212, 2254
F	
\fancyhead	51, 54
\fancyhf	50, 53
\fancypagestyle	52
\filename@area	2187
\filename@parse	2096
\foo	18, 29, 31, 33
\foobar	2935, 2937, 2942, 2943, 2950, 2951
\footskip	1305
\foremargin	1312
\frontmatter	58, 103
G	
\g@addto@macro ..	350, 368, 1729, 3045
\G@refundefinedtrue	776
\gdef	412, 539, 544, 945, 1333, 1682, 1724, 1726
\GetTitleStringDisableCommands	1815
\GetTitleStringExpand	1801
\GetTitleStringNonExpand	1803
\GetTitleStringResult	1806
H	
\hb@xt@	2896
\headheight	1303
\headmargin	1315
\headnamereffalse	3069
\headnamereftrue	3074
\headsep	1304
\hfill	2884, 2896
\hoffset	1296
\hss	2896
I	
\if@filesw ..	710, 1051, 1680, 2471, 2477
\if@safe@actives	889
\ifcase	115, 1195, 1617, 2666, 2725, 2749
\ifcsname	883, 1705, 1730
\ifeof	2111, 2165
\ifetex@unexpanded	267
\ifheadnameref	1907, 1920
\ifin@	317
\ifuatex	1242, 1424
\ifnum	476, 1065, 1170, 1180, 1186, 1243, 1392, 1425, 1547, 1589, 1642, 2057, 2134, 2138, 2143, 2456, 2482, 2546, 2558, 2601, 2614, 2632, 2721, 2745, 2881
\ifodd	124, 2766, 2817
\ifpdf	2452, 2716, 2740
\ifx	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, 2439, 2518, 2523,	
2720, 2744, 2849, 2904, 2911, 3067	
\ifZREF@found	247, 2355, 2362
\ifZREF@immediate	
.	634, 700, 712, 716, 731
\ifZREF@pa@list	1492, 1497
\ifZREF@pl@list	1320, 1325
\ifzref@titleref@expand	1784, 1800
\ifzref@titleref@stripperiod	1797, 1808
\ifZREF@xr@toltxlabel	2231, 2273
\ifZREF@xr@tozreflabel	2217, 2259
\ifZREF@xr@urluse	2101, 2380, 2410
\ifZREF@xr@verbose	2219, 2261, 2282
\ifZREF@xr@zreflabel	
.	2032, 2118, 2132, 2173
\immediate	705, 1681
\in@	314
\IncludeTests	2925, 2960, 3014, 3038
\item	107, 111, 113, 121, 125, 127
K	
\kern	2892
\kv@define@key	644, 655, 666, 671, 688
\kv@key	692, 1611, 1613, 1614, 1628
\kv@parse	689, 1607
\kv@value	690, 1608, 1609, 1616
\kvsetkeys	633, 1832, 2045, 2868
L	
\Label 3043, 3061, 3063, 3066, 3073, 3078	
\label	969, 976, 1840, 2982
\lastxpos	2441
\lastypos	2442
\leavevmode	2871
\List	3042, 3045, 3087
\LogTests	2926, 2961, 3015, 3039
\lst@@caption	1983
\lst@label	1980
\lst@MakeCaption	1979
\LT@c@ption	1971
\ltx@backslashchar 743, 1517, 1567, 2063	
\ltx@car	1611, 2154
\ltx@carthree	3055
\ltx@cdr	1613, 2157
\ltx@empty	290, 499,
565, 632, 719, 796, 1095, 1599,	
1785, 1811, 1818, 1899, 1903,	
1960, 1980, 2093, 2128, 2151,	
2157, 2198, 2352, 2911, 2914, 2917	
\ltx@firstofone	
.	254, 867, 878, 884, 1486, 1487
\ltx@firstoftwo	
.	799, 826, 827, 892, 1067, 2634, 2636
\ltx@firsttwo	2625
\ltx@gobble	
.	250, 355, 390, 623, 663, 969,
970, 976, 1395, 1550, 1630, 1840	
\ltx@gobblethree	977
\ltx@gobbletwo	694, 911, 1008, 1632
\ltx@ifpackageloaded	2305
\ltx@ifUndefined	
.	229, 249, 257, 410, 877, 919,
1103, 1248, 1430, 1452, 1464,	
1483, 1484, 1512, 1555, 1894,	
1896, 2444, 2454, 2480, 2715, 2739	
\ltx@ifundefined	
.	300, 485, 583, 760, 805, 940,
1255, 1274, 1327, 1437, 1499,	
2030, 2290, 2291, 2569, 2640,	
2641, 2642, 2655, 2656, 2657, 2667	
\ltx@LocalAppendToMacro	
.	385, 403, 649, 659, 1565,
2209, 2249, 2252, 2357, 2360,	
2369, 2376, 2382, 2389, 2396, 2408	
\ltx@newif	1320, 1492
\ltx@one	1393, 1548,
.	1736, 2052, 2058, 2722, 2746, 2872
\ltx@onelevel@sanitize	557, 562
\ltx@ReturnAfterFi	2341
\ltx@secondoftwo	311,
.	784, 797, 827, 886, 890, 1069, 2634
\ltx@space	584, 586, 806, 815, 829,
832, 1180, 1186, 1356, 1403,	
1591, 1619, 1662, 2535, 2542, 2605	
\ltx@two	2714, 2720, 2738, 2744
\ltx@undefined	1249, 1431
\ltx@zero	476, 1589,
1642, 1712, 2048, 2482, 2546,	
2558, 2601, 2614, 2632, 2717, 2741	
\luatexversion	1243, 1425
M	
\m@ne	1053
\M@sect	1919
\M@TitleReference	2299
\mag	1285, 2721, 2735, 2745, 2759
\mainmatter	60, 134
\makeatletter	11, 74,
101, 2081, 2928, 2963, 2983, 3041	
\makeatother	16, 99, 2978
\makebox	169, 170
\MakeRobustcommand	232
\marginparsep	1307
\marginparwidth	1306
\mbox	3049
\meaning	2001, 2935, 2937, 2942, 2951
\mediaheight	1291
\mediawidth	1290
\MessageBreak	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, 2446, 2459, 2606, 2618, 2619	

N	
<code>\NeedsTeXFormat</code>	3, 188, 220, 960, 998, 1019, 1039, 1082, 1122, 1213, 1231, 1415, 1578, 1671, 1689, 1772, 2016, 2419, 2431, 2514, 2845, 2900, 2920, 2956, 3010, 3034
<code>\newcommand</code>	18, 78, 85, 91, 167, 968, 975, 981, 1136, 1153, 1154, 1223, 1586, 2044, 2505, 2508, 2530, 2537, 2544, 2556, 2613, 2868, 2964, 2970
<code>\newcount</code>	2047
<code>\newcounter</code>	6, 912, 1009, 1701
<code>\newif</code>	247, 700, 1784, 1797, 2032
<code>\newlabel</code>	2236, 2247, 2278
<code>\newmarks</code>	1595
<code>\newpage</code>	143, 3086
<code>\nfss@text</code>	928
<code>\nofiles</code>	2921
<code>\NR@temp</code>	1902, 1903
<code>\number</code>	94, 109, 915, 920, 954, 1104, 1278, 1279, 1676, 1718, 1744, 1751, 2531, 2533, 2538, 2540, 2545, 2557, 2615, 2627, 2766, 2817, 2874, 3023, 3029
<code>\numexpr</code>	94, 109, 115, 922, 1106, 1167, 1676, 1718, 2193, 2204, 2238, 2323, 2332, 2550, 2562, 2735, 2759, 2881
O	
<code>\oddsidemargin</code>	1299
<code>\on@line</code>	1678, 2128
<code>\openin</code>	2110
P	
<code>\PackageError</code>	258, 269, 294, 304, 489, 1590, 2445, 2457
<code>\PackageInfo</code>	291, 520, 1679, 2117, 2129, 2220, 2262, 2283
<code>\PackageWarning</code>	345, 363, 380, 398, 615, 679, 1620, 1661, 2112
<code>\PackageWarningNoLine</code>	2199, 2318, 2327
<code>\page</code>	2360, 2934, 2943, 2945, 2947
<code>\pagestyle</code>	49
<code>\paperheight</code>	1287
<code>\paperwidth</code>	1286
<code>\par</code>	3046
<code>\part</code>	3067
<code>\pdf@escapehex</code>	1489
<code>\pdf@stricmp</code>	476
<code>\pdf@unescapehex</code>	1490
<code>\pdfhorigin</code>	1266, 1294
<code>\pdflastxpos</code>	2441, 2467
<code>\pdflastypos</code>	2442, 2468
<code>\pdfpageattr</code>	1446, 1454, 1460
<code>\pdfpageheight</code>	1265, 1293, 2666
<code>\pdfpagesattr</code>	1447, 1466, 1472
<code>\pdfpagewidth</code>	1264, 1292
<code>\pdfsavepos</code>	2439, 2440, 2446, 2458, 2472, 2875, 2889
<code>\pdftexversion</code>	2456
<code>\pdfvorigin</code>	1267, 1295
<code>\ProcessOptions</code>	217
<code>\protect</code>	776
<code>\protected</code>	238
<code>\protected@write</code>	715
<code>\providecommand</code>	281, 1674, 2029, 2865, 2866, 2867
<code>\ProvidesPackage</code>	189, 221, 961, 999, 1020, 1040, 1083, 1123, 1214, 1232, 1416, 1579, 1672, 1690, 1773, 2017, 2420, 2432, 2515, 2846, 2901
R	
<code>\read</code>	2163
<code>\refstepcounter</code>	1031
<code>\renewcommand</code>	7, 46, 914
<code>\RequirePackage</code>	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, 2451, 2517, 2522, 2527, 2529, 2848, 2853, 2855, 2903
<code>\reset@font</code>	928
<code>\rightarrow</code>	45
<code>\romannumeral</code>	582, 804, 825, 1600, 2059, 2068, 2069
S	
<code>\savepos</code>	2440
<code>\section</code>	63, 137, 145, 2981
<code>\setcounter</code>	1011
<code>\setlength</code>	2877
<code>\SetupKeyvalOptions</code>	2033
<code>\space</code>	778, 1454, 1466, 2131, 2132, 2136, 2140, 2145, 2446, 2458
<code>\spinemargin</code>	1311
<code>\stepcounter</code>	19, 1013, 1703, 1704
<code>\stockheight</code>	1289
<code>\stockwidth</code>	1288
T	
<code>\tableofcontents</code>	59, 132
<code>\Test</code>	2970, 2986, 2988, 2993, 2995, 2997, 3001, 3002, 3059, 3083
<code>\textbf</code>	2981, 2991, 3004
<code>\textheight</code>	1302
<code>\textit</code>	2981, 2992, 3004
<code>\textwidth</code>	1301
<code>\TeXXeTstate</code>	2482
<code>\the</code>	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,

<code>\ZREF@abspos@marginpar@y@center</code>	<code>\zref@def@extractdefault</code>	855
.....	<code>\ZREF@default</code>	561, 562, 571
<code>\ZREF@abspos@marginpar@y@top</code>	<code>\zref@default</code>	8, 530, 806, 925, 927
.....	<code>\ZREF@delprop</code>
<code>\ZREF@abspos@media@height</code>	412, 415, 417, 452, 455, 457
.....	<code>\zref@delprop</code>	411, 451
<code>\ZREF@abspos@media@width</code>	<code>\ZREF@df@dot</code>	2863, 2867, 2896
.....	<code>\ZREF@df@min</code>	2860, 2865, 2883
<code>\ZREF@abspos@media@x@center</code>	<code>\ZREF@df@unit</code>	2857, 2866, 2877
.....	<code>\ZREF@dotfill</code>	2879, 2886, 2895
<code>\ZREF@abspos@media@x@left</code>	<code>\ZREF@ENV@line</code>	2914, 2917
.....	<code>\zref@env@line</code>	2909, 2910
<code>\ZREF@abspos@media@x@right</code>	<code>\ZREF@extract</code>	803, 820, 823, 875
.....	<code>\zref@extract</code>
<code>\ZREF@abspos@media@y@bottom</code>	8, 95, 96, 109, 140, 803, 823, 852, 870, 875, 988, 1109, 1205, 1357, 1404, 1405, 1529, 1559, 1747, 1748, 1841, 2506, 2509, 2681, 2685, 2688, 2694, 2700, 2703, 2708, 2712, 2727, 2751, 2765, 2767, 2769, 2774, 2778, 2783, 2784, 2788, 2792, 2799, 2803, 2807, 2814, 2819, 2822, 2823, 2828, 2832, 2997, 3001, 3002, 3054
<code>\ZREF@abspos@media@y@center</code>	<code>\ZREF@extractdefault</code>	824, 840, 843, 874
.....	<code>\zref@extractdefault</code>
<code>\ZREF@abspos@media@y@top</code>	7, 116, 117, 816, 843, 863, 869, 874, 1065, 1066, 1163, 1178, 1224, 1750, 2292, 2295, 2296, 2300, 2301, 2304, 2306, 2307, 2309, 2311, 2533, 2540, 2599, 2627, 2766, 2817, 2993, 2995
.....	<code>\ZREF@false</code>	676, 686
<code>\ZREF@abspos@stock@x@center</code>	<code>\ZREF@foundfalse</code>	2353
.....	<code>\ZREF@foundtrue</code>	2400
<code>\ZREF@abspos@stock@x@left</code>	<code>\ZREF@getcurrent</code>	581, 592, 595, 873
.....	<code>\zref@getcurrent</code>
<code>\ZREF@abspos@stock@x@right</code>	7, 595, 868, 873, 2986, 2988
.....	<code>\zref@hex</code>	1460, 1472, 1486, 1489
<code>\ZREF@abspos@stock@y@bottom</code>	<code>\zref@ifabsposnumundefined</code>	2626, 2631
.....	<code>\zref@ifabsposundefined</code>	2624
<code>\ZREF@abspos@stock@y@center</code>	<code>\ZREF@ifDefinable</code>	241, 762, 990, 993, 1072, 1114, 1150, 1321, 1493, 1710, 1764, 1831, 1834, 2075, 2490, 2495, 2500, 2869
.....	<code>\ZREF@iflastpage</code>	1073, 1075, 1075
<code>\ZREF@abspos@stock@y@top</code>	<code>\zref@iflastpage</code>	12, 1064, 1078
.....	<code>\zref@iflistcontainsprop</code>	6, 309, 344, 362, 379, 397, 647, 657
<code>\ZREF@abspos@used</code>	<code>\zref@iflistundefined</code>	6, 288, 299, 303, 310
.....	<code>\zref@ifpropundefined</code>	7, 484, 488, 518, 547, 614, 827, 1400, 1641, 2347
<code>\ZREF@absposnum</code>	<code>\ZREF@ifrefcontainsprop</code>	786, 794
.....	<code>\zref@ifrefcontainsprop</code>
<code>\zref@absposnumused</code>	8, 782, 1402, 2308, 2404, 2405
.....	<code>\ZREF@ifrefundefined</code>
<code>\zref@absposnumx</code>	764, 767, 1160, 1171, 1181
.....	<code>\zref@ifrefundefined</code>	8, 759, 769, 775, 783, 826, 1172, 1355, 1541, 1740, 2547, 2559, 2595, 2625, 2633, 2878, 2999, 3051
<code>\zref@absposnumy</code>	<code>\ZREF@immediatefalse</code>	677
.....		
<code>\zref@absposused</code>		
.....		
<code>\zref@absposx</code>		
.....		
<code>\zref@absposy</code>		
.....		
<code>\zref@addprop</code>		
.....		
<code>\zref@addprops</code>		
.....		
<code>\ZREF@addtoks</code>		
.....		
<code>\ZREF@base@ok</code>		
.....		
<code>\ZREF@call</code>		
.....		
<code>\ZREF@def@abspos</code>		
.....		
<code>\zref@def@absposnumx</code>		
.....		
<code>\ZREF@def@absposnumy</code>		
.....		
<code>\zref@def@absposnumy</code>		
.....		
<code>\zref@def@absposx</code>		
.....		
<code>\zref@def@absposy</code>		
.....		
<code>\ZREF@def@extract</code>		
.....		
<code>\zref@def@extract</code>		
.....		
<code>\ZREF@def@extractdefault</code>		
.....		

\ZREF@immediatetrue 674, 703
\ZREF@label 602, 626, 636, 639, 709, 1059
\zref@label 7, 596, 972, 3044
\zref@labelbykv 629, 979
\zref@labelbylist
. 7, 597, 599, 1100, 1739, 2492
\zref@labelbyprops 7, 88,
609, 1158, 2497, 2502, 2876, 2890
\zref@listexists 6, 302, 321,
341, 360, 376, 395, 418, 458, 601
\zref@listforloop 320, 656
\zref@listpageattr 1493
\zref@listpagelayout 1321
\zref@localaddprop 394, 2941, 2944
\zref@localaddprops 375
\zref@localdelprop 414, 454, 668
\ZREF@mainlist 597, 931,
934, 937, 1016, 1028, 1783, 2428
\ZREF@makeperpage@opt 1712, 1715
\ZREF@MARKS@DefineProp
. 1604, 1605, 1606, 1640
\zref@marks@register
. 1586, 1591, 1623, 1662
\ZREF@name 228, 258, 269,
291, 294, 304, 345, 363, 380,
398, 489, 508, 520, 524, 548,
563, 615, 679, 691, 1590, 2445, 2457
\ZREF@NAME@bot 1619, 1639
\ZREF@NAME@first 1618, 1638
\ZREF@NAME@top 1617, 1637
\zref@newlabel
. 7, 281, 284, 749, 2191, 2277
\zref@newlist 6, 287, 934,
1049, 1092, 1614, 1699, 2466, 2938
\ZREF@newprop 497, 500, 503
\zref@newprop 6, 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,
2467, 2468, 2908, 2909, 2936, 2984
\ZREF@NewPropAnchor 938, 2077, 2427
\ZREF@NewPropPageValue
. 953, 1094, 1698
\ZREF@NewPropTheotype 949, 2248
\ZREF@NewPropTitle 944, 1782, 2078
\ZREF@nextpage 1151, 1155
\ZREF@nil 544, 796, 835, 2164, 2170,
2176, 2181, 2191, 2207, 2236,
2244, 2335, 2342, 2351, 2354, 2393
\ZREF@NOVALUE 802
\ZREF@novalue 795, 796, 802
\ZREF@np@call@next 1145, 1149, 1204
\ZREF@np@call@nonext 1142, 1148, 1200
\ZREF@np@call@unknown
. 1138, 1147, 1196
\ZREF@np@setup@i 1137, 1140
\ZREF@np@setup@ii 1141, 1144
\ZREF@number
. 919, 1529, 1533, 1588, 2632
\ZREF@org@@@begintheorem 2008
\ZREF@org@@@caption 1848
\ZREF@org@@@chapter 1860, 1916
\ZREF@org@@@opargbegintheorem 1993
\ZREF@org@@@part 1854
\ZREF@org@@@schapter 1878
\ZREF@org@@@sect 1866
\ZREF@org@@@spart 1872
\ZREF@org@@@ssect 1884
\ZREF@org@beamer@section 1934
\ZREF@org@beamer@subsection 1940
\ZREF@org@beamer@subsubsection 1946
\ZREF@org@descriptionlabel 1890
\ZREF@org@lst@MakeCaption 1986
\ZREF@org@LT@ccOption 1972
\ZREF@org@M@sect 1925
\ZREF@org@refstepcounter 1033
\ZREF@org@stepcounter 1703, 1708
\ZREF@org@testdef
. 1328, 1330, 1500, 1502
\ZREF@org@thepage 713, 717
\ZREF@org@ttl@sect@i 1954
\ZREF@org@ttl@straight@i 1965
\ZREF@org@write 704, 705
\ZREF@P 504,
505, 507, 509, 518, 521, 525,
535, 536, 538, 539, 540, 544,
720, 724, 725, 734, 738, 743, 744
\ZREF@pa@AfterLastShipout 1496, 1574
\ZREF@pa@AtVeryEnd 1509, 1512, 1565
\ZREF@pa@ListPage 1519, 1535
\ZREF@pa@listtrue 1494
\ZREF@page@max 1326, 1392, 1498, 1547
\zref@pageattr 1527
\zref@pageattr@used 1532
\ZREF@pagenum@last 1177, 1180
\ZREF@pagenum@this
. 1162, 1167, 1170, 1180, 1186
\ZREF@par 507, 532
\ZREF@param
. 421, 422, 441, 459, 476, 645,
646, 647, 651, 672, 673, 676, 681
\ZREF@patch 248, 1030, 1845, 1851,
1857, 1863, 1869, 1875, 1881,
1887, 1918, 1931, 1937, 1943,
1951, 1957, 1970, 1978, 1990, 2005
\zref@pdfpageattr 1456, 1542
\zref@pdfpageattr@used 1457
\zref@pdfpagesattr 1468, 1557, 1568
\zref@pdfpagesattr@used 1469, 1562
\ZREF@pl@AfterLastShipout 1324, 1411
\ZREF@pl@AtVeryEnd 1337, 1340
\ZREF@pl@ListEntry
. 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
\ZREF@pl@ListPage 1344, 1350
\ZREF@pl@listtrue 1322
\zref@pos@label@used 2605
\zref@pos@num@used 2619
\zref@prop 323, 331, 332, 336, 657, 661

`\zref@propexists` 7, 343, 361, 378, 396, 487, 577, 646, 667, 982
`\ZREF@refname@next` 1165, 1172, 1181, 1205
`\ZREF@refname@this` 1157, 1158, 1160, 1163
`\ZREF@RefPrefix` . . 283, 285, 1332, 1504
`\ZREF@refused` 772, 774
`\zref@refused` 8, 768, 771, 848, 859, 987, 994, 1076, 1077, 1112, 1227, 1533, 1563, 1839, 2594, 2602, 2615, 3050
`\zref@require@unique` 10, 908, 1702, 2854
`\ZREF@Robust` 231, 237, 243, 284, 287, 302, 309, 340, 359, 375, 394, 411, 414, 451, 454, 487, 494, 546, 576, 596, 599, 609, 629, 701, 771, 844, 855, 866, 882, 908, 924, 930, 1111, 1532, 1562, 1786, 1799, 2575, 2578, 2581, 2584, 2590
`\ZREF@SavedEscapechar` 460, 467
`\zref@savepos` 18, 2470, 2478, 2484
`\ZREF@savepos@ok` 2511, 2523
`\zref@setcurrent` 6, 81, 540, 576, 696, 1032
`\zref@setdefault` 8, 924, 927
`\zref@setmainlist` 9, 930
`\zref@showprop` 546
`\ZREF@STAR` 1612, 1636
`\ZREF@stripperperiod` 1810, 1818
`\ZREF@temp` 193, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 232, 233, 439, 440, 441, 719, 740, 741, 749, 1247, 1264, 1265, 1266, 1267, 1271, 1285, 1286, 1287, 1288, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1297, 1298, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1331, 1332, 1429, 1446, 1447, 1451, 1462, 1474, 1476, 1482, 1483, 1484, 1485, 1503, 1504, 1611, 1612, 1959, 1960, 2051, 2061, 2064, 2068, 2598, 2601, 2602, 2714, 2717, 2720, 2722, 2725, 2738, 2741, 2744, 2746, 2749, 2874
`\ZREF@TempName` 1587, 1599, 1600, 1602, 1628, 1641, 1645, 1653, 1664
`\ZREF@TempNum` 1588, 1589, 1593, 1600, 1642, 1655
`\zref@thepage` 13, 1108, 1117
`\zref@thepage@atbegshi@hook` 1095, 1099
`\zref@thepage@name` 13, 1103, 1109, 1112, 1166
`\zref@thepage@refused` 1111, 1116
`\ZREF@titleref` 1835, 1837
`\zref@titleref@cleanup` 1786, 1826
`\zref@titleref@current` 944, 1805, 1809, 1810, 1829
`\ZREF@titleref@hook` 1785, 1789, 1793, 1816
`\zref@titleref@setcurrent` 1799, 1847, 1853, 1859, 1865, 1871, 1877, 1883, 1889, 1897, 1900, 1904, 1908, 1910, 1921, 1923, 1933, 1939, 1945, 1953, 1961, 1963, 1973, 1982, 1992, 2007
`\zref@titleref@stripperperiodtrue` . . . 1798
`\ZREF@true` 673, 687
`\ZREF@u@getcurrent` 590
`\zref@unhex` . . . 1487, 1490, 1528, 1558
`\ZREF@UpdatePdfTeX` 246, 2448, 2461
`\ZREF@value` 556, 557, 570
`\ZREF@wrapper@babel` 899, 905
`\zref@wrapper@babel` 10, 140, 764, 772, 845, 856, 882, 972, 979, 983, 1073, 1835, 2576, 2579, 2582, 2585, 2591
`\zref@wrapper@immediate` 10, 87, 635, 701, 1058
`\ZREF@wrapper@unexpanded` . 866, 880
`\zref@wrapper@unexpanded` 10, 867, 872, 877, 2287, 2972
`\ZREF@wu@extract` 818, 870
`\ZREF@wu@extractdefault` . . . 838, 869
`\ZREF@wu@getcurrent` 590, 868
`\ZREF@X` 496, 499, 536
`\zref@xr@` 2041
`\ZREF@xr@@AddUrl` 2053, 2056
`\ZREF@xr@@input` 2184, 2279
`\ZREF@xr@AddURL` . . 2049, 2102, 2381
`\ZREF@xr@checkfile` . . 2106, 2109, 2159
`\ZREF@xr@checkkey` 2337, 2346
`\ZREF@xr@checklist` 2207, 2335
`\zref@xr@ext` 19, 2029, 2097
`\ZREF@xr@externaldocument` 2084, 2087, 2091
`\ZREF@xr@externalfile` 2094, 2095, 2213, 2255
`\ZREF@xr@file` . . 2095, 2110, 2113, 2119, 2130, 2153, 2201, 2320, 2329
`\ZREF@xr@filelist` 2093, 2151, 2154, 2156, 2157, 2185, 2186
`\ZREF@xr@found` 2121, 2131, 2193, 2238
`\ZREF@xr@graburl` 2097, 2099
`\ZREF@xr@ignored@empty` 2122, 2134, 2136, 2203, 2204
`\ZREF@xr@ignored@ltx` 2124, 2143, 2145, 2331, 2332
`\ZREF@xr@ignored@zref` 2123, 2138, 2140, 2322, 2323
`\ZREF@xr@line` . . 2163, 2164, 2176, 2181
`\ZREF@xr@list` 2197, 2198
`\ZREF@xr@ltx@ignorewarning` . . . 2326
`\ZREF@xr@newlabel` 2179, 2278
`\ZREF@xr@prefix` 2092, 2192, 2228, 2232, 2237, 2263, 2270, 2274
`\ZREF@xr@process@label` . . 2181, 2236
`\ZREF@xr@process@zreflabel` 2176, 2191

<code>\ZREF@xr@processfile</code>	2109 , 2162	<code>\ZREF@xr@zref@ignorewarning</code>	2228 , 2270 , 2317
<code>\ZREF@xr@processline</code>	2164 , 2170	<code>\ZREF@xr@zref@newlabel</code>	..	2174 , 2277
<code>\ZREF@xr@refname</code>	2192 , 2218 , 2225 , 2237 , 2260 , 2267	<code>\ZREF@xr@zref@labelfalse</code>	2083
<code>\ZREF@xr@relax</code>	2280 , 2367	<code>\ZREF@xr@zref@labeltrue</code>	2086
<code>\ZREF@xr@scanparams</code>	2242 , 2351	<code>\ZREF@zref</code>	983 , 986
<code>\ZREF@xr@scantitleref</code>	2354 , 2393	<code>\ZREF@zsavepos</code>	2475 , 2492 , 2497 , 2502	
<code>\ZREF@xr@temp</code>	2366 , 2367	<code>\zrefused</code>	..	11 , 92 , 93 , 161 , 162 , 163 , 993
<code>\ZREF@xr@tempname</code>	2195 , 2196 , 2216 , 2221 , 2232 , 2240 , 2241 , 2258 , 2274	<code>\zruns</code>	16 , 1674 , 3018 , 3023 , 3029
<code>\ZREF@xr@temprefname</code>	2196 , 2208 , 2210 , 2226 , 2241 , 2243 , 2250 , 2253 , 2268	<code>\zsavepos</code>	17 , 157 , 158 , 2490
<code>\ZREF@xr@theURL</code>	2059 , 2061 , 2063 , 2069 , 2104 , 2385	<code>\zsaveposx</code>	17 , 2495
<code>\ZREF@xr@tolabel</code>	..	2232 , 2274 , 2281	<code>\zsaveposy</code>	2500
<code>\ZREF@xr@URL</code>	2047 , 2057 , 2058 , 2059		<code>\zthepage</code>	13 , 1114
<code>\ZREF@xr@url</code>	..	2100 , 2102 , 2103 , 2411	<code>\ztitleref</code>	17 , 1834 , 3047
<code>\ZREF@xr@urlcheck</code>	..	2216 , 2258 , 2403	<code>\ztitlerefsetup</code>	17 , 1819
			<code>\ztotpages</code>	15 , 124 , 1223
			<code>\zunknownnextpagename</code>	14 , 1154 , 1197	
			<code>\zunmakeperpage</code>	16 , 1764
			<code>\zxrsetup</code>	19 , 2044