The zref package

Heiko Oberdiek*

2025-06-08 v2.36

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 user . . . 10
- 3.2 Module abspage . . . 11
- 3.3 Module lastpage . . . 12
  - 3.3.1 Tests for last page . . . 12
  - 3.3.2 Example . . . 12
- 3.4 Module thepage . . . 13
- 3.5 Module nextpage . . . 13
  - 3.5.1 Configuration . . . 14
  - 3.5.2 Example . . . 14
- 3.6 Module totpages . . . 15
- 3.7 Module pagelayout . . . 15
- 3.8 Module marks . . . 16
- 3.9 Module runs . . . 16
- 3.10 Module perpage . . . 16
- 3.11 Module counter . . . 17
- 3.12 Module titleref . . . 17
- 3.13 Module savepos . . . 17
- 3.14 Module abspos . . . 18

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

3.15 Module dotfill . . . 19
3.16 Module env . . . 19
3.17 Module xr . . . 19
3.18 Module pageattr . . . 20

4 ToDo **20**

5 Example **20**

6 Implementation **23**
6.1 Package zref . . . 23
6.1.1 Identification . . . 23
6.1.2 Load basic module . . . 23
6.1.3 Process options . . . 23
6.2 Module base . . . 23
6.2.1 Prefixes . . . 23
6.2.2 Identification . . . 24
6.2.3 Utilities . . . 24
6.2.4 Check for ε-TEX . . . 25
6.2.5 Auxiliary file stuff . . . 25
6.2.6 Property lists . . . 26
6.2.7 Properties . . . 30
6.2.8 Reference generation . . . 32
6.2.9 Reference querying and extracting . . . 35
6.2.10 Compatibility with babel . . . 38
6.2.11 Unique counter support . . . 39
6.2.12 Utilities . . . 39
6.2.13 Setup . . . 39
6.3 Module user . . . 40
6.4 Module abspage . . . 41
6.5 Module counter . . . 42
6.6 Module lastpage . . . 42
6.7 Module thepage . . . 44
6.8 Module nextpage . . . 44
6.9 Module totpages . . . 46
6.10 Module pagelayout . . . 46
6.10.1 Define layout properties . . . 47
6.11 Module pageattr . . . 50
6.12 Module marks . . . 53
6.13 Module runs . . . 54
6.14 Module perpage . . . 55
6.15 Module titleref . . . 57
6.15.1 Implementation . . . 57
6.15.2 User interface . . . 59
6.15.3 Patches for section and caption commands . . . 59
6.15.4 Environment description . . . 60
6.15.5 Class memoir . . . 60
6.15.6 Class beamer . . . 61
6.15.7 Package titlesec . . . 61
6.15.8 Package longtable . . . 61
6.15.9 Package listings . . . 62
6.15.10 Theorems . . . 62
6.16 Module xr . . . 62
6.17 Module hyperref . . . 70
6.18 Module savepos . . . 70
6.18.1 Identification . . . 70
6.18.2 Availability . . . 70
6.18.3 Setup . . . 70

6.18.4 User macros . . . 70
6.19 Module abspos . . . 72
6.19.1 Identification . . . 72
6.19.2 Media . . . 74
6.19.3 Paper . . . 75
6.19.4 Origin . . . 76
6.19.5 Header . . . 77
6.19.6 Body . . . 78
6.19.7 Footer . . . 78
6.19.8 Marginal notes . . . 79
6.19.9 Stock paper . . . 79
6.20 Module dotfill . . . 80
6.21 Module env . . . 81

7 Installation **81**
7.1 Download . . . 81
7.2 Bundle installation . . . 82
7.3 Package installation . . . 82
7.4 Refresh file name databases . . . 82
7.5 Some details for the interested . . . 82

8 References **83**

9 History **83**
[2006/02/20 v1.0] . . . 83
[2006/05/03 v1.1] . . . 83
[2006/05/25 v1.2] . . . 83
[2006/09/08 v1.3] . . . 84
[2007/01/23 v1.4] . . . 84
[2007/02/18 v1.5] . . . 84
[2007/04/06 v1.6] . . . 84
[2007/04/17 v1.7] . . . 84
[2007/04/22 v1.8] . . . 84
[2007/05/02 v1.9] . . . 84
[2007/05/06 v2.0] . . . 84
[2007/05/28 v2.1] . . . 84
[2008/09/21 v2.2] . . . 84
[2008/10/01 v2.3] . . . 84
[2009/08/07 v2.4] . . . 85
[2009/12/06 v2.5] . . . 85
[2009/12/07 v2.6] . . . 85
[2009/12/08 v2.7] . . . 85
[2010/03/26 v2.8] . . . 85
[2010/03/29 v2.9] . . . 85
[2010/04/08 v2.10] . . . 85
[2010/04/15 v2.11] . . . 85
[2010/04/17 v2.12] . . . 85
[2010/04/19 v2.13] . . . 86
[2010/04/22 v2.14] . . . 86
[2010/04/23 v2.15] . . . 86
[2010/04/28 v2.16] . . . 86
[2010/05/01 v2.17] . . . 86
[2010/05/13 v2.18] . . . 86
[2010/10/22 v2.19] . . . 87
[2011/02/12 v2.20] . . . 87
[2011/03/18 v2.21] . . . 87
[2011/10/05 v2.22] . . . 87
[2011/12/05 v2.23] . . . 87

[2012/04/04 v2.24] . . . 87
[2016/05/16 v2.25] . . . 87
[2016/05/21 v2.26] . . . 87
[2018/11/21 v2.27] . . . 87
[2019/11/29 v2.28] . . . 87
[2020-03-03 v2.29] . . . 88
[2020-03-04 v2.30] . . . 88
[2020-05-28 v2.31] . . . 88
[2020-07-03 v2.32] . . . 88
[2022-03-08 v2.33] . . . 88
[2022-04-07 v2.34] . . . 88
[2023-09-14 v.2.35] . . . 88
[2025-06-08 v2.36] . . . 88

10 Index **89**

1 Introduction

Standard LaTeX'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 LaTeX 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 LaTeX behaviour

References are created by the `\label` command:

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

Now LaTeX 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 if fixed in the LaTeX 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 LaTeX and other packages.

titleref stores its title data into the first entry in the list. LaTeX 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 LaTeX3 that will need some time before its first release. Thus I have implemented it as LaTeX 2$_\varepsilon$ package without disturbing the existing LaTeX 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@iflistundefined`exp {⟨*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 LaTeX's `\currentlabel`.

`\zref@getcurrent`$^{\text{exp2}}$ {⟨*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@ifpropundefined`$^{\text{exp}}$ {⟨*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@extractdefault`$^{\text{exp2}}$ {⟨*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@extract`$^{\text{exp2}}$ {⟨*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@ifrefundefined`$^{\text{exp}}$ {⟨*refname*⟩} {⟨*then*⟩} {⟨*else*⟩}

Macro `\zref@ifrefundefined` calls arguments ⟨*then*⟩ or ⟨*else*⟩ dependent on the existence of the reference ⟨*refname*⟩.

`\zifrefundefined` {⟨*refname*⟩} {⟨*then*⟩} {⟨*else*⟩}

Macro `\zifrefundefined` calls `\ref@refused` before executing `\zref@ifrefundefined`. Babel shorthands are supported in ⟨*refname*⟩.

`\zref@ifrefcontainsprop`$^{\text{exp}}$ {⟨*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*⟩
	`theotype`		⟨*empty*⟩
	`title`		⟨*empty*⟩
	`url`		⟨*empty*⟩

[1] Module 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 LaTeX 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`:

$$\texttt{\textbackslash zref\{}x\texttt{\}} \equiv \texttt{\textbackslash zref[default]\{}x\texttt{\}}$$

`\zpageref` {⟨*refname*⟩}[babel]

Convenience macro, similar to `\pageref`.

$$\texttt{\textbackslash zpageref\{}x\texttt{\}} \equiv \texttt{\textbackslash zref[page]\{}x\texttt{\}}$$

`\zrefused` {⟨*refname*⟩}[babel]

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

3.2 Module abspage

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

Thus you can reference the absolute page number:

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

The example also makes use of module lastpage.

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

3.3 Module lastpage

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

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

or

`\zpageref{LastPage}` (module user)

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

3.3.1 Tests for last page

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

`\zref@iflastpage`$^{\text{exp}}$ {⟨*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

```
⟨*example-lastpage⟩
%<<END_EXAMPLE
\NeedsTeXFormat{LaTeX2e}
\documentclass{report}

\newcounter{foo}
\renewcommand*{\thefoo}{\Alph{foo}}

\usepackage{zref-lastpage,zref-user}[2019/11/29]

\makeatletter
\zref@newprop{thefoo}{\thefoo}
\zref@newprop{valuefoo}{\the\value{foo}}
\zref@newprop{chapter}{\thechapter}
\zref@addprops{LastPage}{thefoo,valuefoo,chapter}
\makeatother

\newcommand*{\foo}{%
```

```
  \stepcounter{foo}%
  [Current foo: \thefoo]%
}

\begin{document}
  \chapter{First chapter}
  Last page is \zref{LastPage}.\\
  Last chapter is \zref[chapter]{LastPage}.\\
  Last foo is \zref[thefoo]{LastPage}.\\
  Last value of foo is \zref[valuefoo]{LastPage}.\\
  \foo
  \chapter{Second chapter}
  \foo\foo\foo
  \chapter{Last chapter}
  \foo
\end{document}
%END_EXAMPLE
```
⟨/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@name`$^{\text{exp}}$ {⟨*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@thepage`$^{\text{exp}}$ {⟨*absolute page number*⟩}
`\zref@thepage@refused` {⟨*absolute page number*⟩}

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

3.5 Module nextpage

`\znextpage`

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

1. The next page is not known yet because of undefined references. Then `\zunknownnextpagename` is used instead. The default for this macro is the default of property page.
2. This page is the last page. Then `\znonextpagename` is used. Its default is empty.

3. The next page is known, then `\thepage` of the next page is used (the value of property `page` of the next page).

3.5.1 Configuration

The behaviour can be configured by the following macros.

`\zunknownnextpagename`
`\znonextpagename`

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

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

`\znextpagesetup` {⟨*unknown*⟩} {⟨*no next*⟩} {⟨*next*⟩}

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

```
⟨*example-nextpage⟩
%<<END_EXAMPLE
\documentclass{book}

\usepackage{zref-nextpage}[2019/11/29]
\znextpagesetup
  {\thepage}% next page is unknown
  {\thepage\ (#1)}% this page is last page
  {\thepage\ $\rightarrow$ #1}% next page is known
\renewcommand*{\znonextpagename}{last page}

\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\fancyhead[LE,RO]{\znextpage}
\fancypagestyle{plain}{%
  \fancyhf{}%
  \fancyhead[LE,RO]{\znextpage}%
}

\begin{document}
\frontmatter
  \tableofcontents
\mainmatter
  \chapter{Hello World}
  \clearpage
  \section{Last section}
\end{document}
%END_EXAMPLE
⟨/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.

`\ztotpages`$^{\text{exp}}$

Prints the total number of pages or `0` if this number is not yet known. It expands to an explicit number and can also 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
`\mag`	mag	
`\paperwidth`	paperwidth	
`\paperheight`	paperheight	
`\stockwidth`	stockwidth	class memoir
`\stockheight`	stockheight	class memoir
`\pdfpagewidth`	pdfpagewidth	pdfTEX, LuaTEX
`\pdfpageheight`	pdfpageheight	pdfTEX, LuaTEX
`\pdfhorigin`	pdfhorigin	pdfTEX, LuaTEX
`\pdfvorigin`	pdfvorigin	pdfTEX, LuaTEX
`\hoffset`	hoffset	
`\voffset`	voffset	
`\topmargin`	topmargin	
`\oddsidemargin`	oddsidemargin	
`\evensidemargin`	evensidemargin	
`\textwidth`	textwidth	
`\textheight`	textheight	
`\headheight`	headheight	
`\headsep`	headsep	
`\footskip`	footskip	
`\marginparwidth`	marginparwidth	
`\marginparsep`	marginparsep	
`\columnwidth`	columnwidth	
`\columnsep`	columnsep	

`\zlistpagelayout`

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

3.8 Module marks

ToDo.

3.9 Module runs

Module runs counts the LaTeX runs since last .aux file creation and prints the number in the .log file.

\zruns^{exp}

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 reimplementation of the autoref feature, see the section 4 with the todo list.

3.12 Module titleref

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

`\ztitleref` {⟨*refname*⟩}[babel]

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

`\ztitlerefsetup` {*key*$_1$=*value*$_1$, *key*$_2$=*value*$_2$, ...}

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

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

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

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

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

3.13 Module savepos

This option supports a feature that all engines nowadays provide: (and XeTeX). They are able to tell the current position on the page. The page position is not instantly known. First the page must be constructed by 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.

`\zposx`$^{\text{exp}}$ {⟨*refname*⟩}
`\zposy`$^{\text{exp}}$ {⟨*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 LATEX that the reference is actually used (the notifying is not expandable). Therefore you should mark the reference as used by `\zrefused`.

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

`\zref@savepos`

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

Thus `\zsavepos` is basically `\zref@savepos` followed by `\zref@labelbylist{`⟨*refname*⟩`}{savepos}` `\savepos` also adds `\zref@savepos` at the end to support `\beginR` where the whatits are processed in reverse order. The property list `savepos` contains the properties `posx` and `posy`.

3.14 Module **abspos**

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

`\zref@absposx{`⟨*label*⟩`}{`⟨*value*⟩`}{`⟨*position*⟩`}`
`\zref@absposy{`⟨*label*⟩`}{`⟨*value*⟩`}{`⟨*position*⟩`}`

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

⟨*label*⟩ is a label set with `\zlabel` or `\zsavepos` that allows to retrieve the absolute page number.

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

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

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

3.15 Module dotfill

`\zdotfill`

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

`\zdotfillsetup {`*key*$_1$`=`*value*$_1$`, `*key*$_2$`=`*value*$_2$`, ...}`

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

`min=`⟨*count value*⟩
: If the actual number of dots are smaller than ⟨*count value*⟩, then the dots are suppressed. Default: `2`.

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

`dot=`⟨*value*⟩
: The dot itself is given by ⟨*value*⟩. Default: `.` (dot, same as the dot in `\dotfill`).

3.16 Module env

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

3.17 Module xr

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

`\zexternaldocument*[`⟨*prefix*⟩`]`$^{\text{babel}}$ `{`⟨*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@newlabel`s 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 {`*key*$_1$`=`*value*$_1$`, `*key*$_2$`=`*value*$_2$`, ...}`

The following setup options are available:

ext: It sets the default extension.

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

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

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

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

`\zref@xr@ext`

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

3.18 Module pageattr

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

`\zref@pdfpageattr{`⟨*absolute page number*⟩`}`

`\zref@pdfpagesattr{`⟨*absolute page number*⟩`}`

4 ToDo

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

- Other applications: autoref, hyperref, ...

5 Example

```
⟨*example⟩
\documentclass{book}

\usepackage[ngerman]{babel}%

\usepackage[savepos,totpages,titleref,dotfill,counter,user]{zref}

```

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

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

```
\makeatletter
\zref@newprop{chaptitle}{}
\zref@addprop{main}{chaptitle}

\newcommand*{\ChapterStart}[2]{%
```

```
  \cleardoublepage
  \def\current@chapid{#1}%
  \zref@setcurrent{chaptitle}{#2}%
  \chapter{#2}%
  \zlabel{chap:#1}%
}
\newcommand*{\ChapterStop}{%
  \cleardoublepage
  \zref@wrapper@immediate{%
    \zref@labelbyprops{chapend:\current@chapid}{abspage}%
  }%
}
```

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

```
\newcommand*{\ChapterPages}[1]{%
  \zrefused{chap:#1}%
  \zrefused{chapend:#1}%
  \number\numexpr
    \zref@extract{chapend:#1}{abspage}%
    -\zref@extract{chap:#1}{abspage}%
    +1\relax
}
\makeatother
\begin{document}
```

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

```
\makeatletter

\frontmatter
\zlabel{documentstart}

\begin{itemize}
\item
  The frontmatter part has
  \number\numexpr\zref@extract{chap:first}{abspage}-1\relax
  ~pages.
\item
  Chapter \zref{chap:first} has \ChapterPages{first} page(s).
\item
  Section \zref{hello} is on the
  \ifcase\numexpr
    \zref@extractdefault{hello}{page}{0}%
    -\zref@extractdefault{chap:first}{page}{0}%
    +1\relax
    ??\or first\or second\or third\or forth\fi
  ~page inside its chapter.
\item
  The document has
  \zref[abspage]{LastPage} pages.
  This number is \ifodd\ztotpages odd\else even\fi.
\item
  The last page is labeled with \zpageref{LastPage}.
\item
  The title of chapter \zref{chap:next} %
  is ``\zref[chaptitle]{chap:next}''.
\end{itemize}

\tableofcontents

\mainmatter
\ChapterStart{first}{First chapter}
```

```
136
```
The user level commands should protect babel shorthands where possible. On the other side, expandable extracting macros are useful in calculations, see above the examples with `\numexpr`.
```
137 \section{Test}
138 \zlabel{a"o}
139 Section \zref{a"o} on page
140 \zref@wrapper@babel\zref@extract{a"o}{page}.
141
142 Text.
143 \newpage
144
145 \section{Hello World}
146 \zlabel{hello}
147
148 \ChapterStop
149
150 \ChapterStart{next}{Next chapter with \emph{umlauts}: "a"o"u"s}
151
```
Here an example follows that makes use of the "savepos" feature. The position on the page is not known before the page is constructed and shipped out. Therefore the position ist stored in references and are available for calculations in the next 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}
```

```
⟨/example⟩
```

6 Implementation

6.1 Package zref

6.1.1 Identification

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref}
  [2025-06-08 v2.36 A new reference scheme for LaTeX (HO)]%
```

6.1.2 Load basic module

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

Abort package loading if zref-base could not be loaded successfully.

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

6.1.3 Process options

Known modules are loaded and the release date is checked.

```
\def\ZREF@temp#1{%
  \DeclareOption{#1}{%
    \AtEndOfPackage{%
      \RequirePackage{zref-#1}[2019/11/29]%
    }%
  }%
}
\ZREF@temp{abspage}
\ZREF@temp{counter}
\ZREF@temp{dotfill}
\ZREF@temp{hyperref}
\ZREF@temp{lastpage}
\ZREF@temp{marks}
\ZREF@temp{nextpage}
\ZREF@temp{pageattr}
\ZREF@temp{pagelayout}
\ZREF@temp{perpage}
\ZREF@temp{runs}
\ZREF@temp{savepos}
\ZREF@temp{thepage}
\ZREF@temp{titleref}
\ZREF@temp{totpages}
\ZREF@temp{user}
\ZREF@temp{xr}
\ProcessOptions\relax
⟨/package⟩
```

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:

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

\ZREF@: Internal macros.

\Z@L@*listname*: The properties of the list ⟨*listname*⟩.

\Z@D@*propname*: The default value for property ⟨*propname*⟩.

\Z@E@*propname*: Extract function for property ⟨*propname*⟩.

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

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

`\Z@R@`*labelname*`:` Data for reference ⟨*labelname*⟩.

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

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

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

`ZREF@TR:` Setup for module titleref.

6.2.2 Identification

```
⟨*base⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-base}%
  [2025-06-08 v2.36 Module base for zref (HO)]%
```

6.2.3 Utilities

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

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

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

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

`\ZREF@Robust`

```
  \def\ZREF@Robust#1#2{%
    \def\ZREF@temp{\MakeRobustcommand#2}%
    \afterassignment\ZREF@temp
    #1#2%
  }%

}{%
```

`\ZREF@Robust`

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

}
```

`\ZREF@IfDefinable`

```
\def\ZREF@IfDefinable#1#2#3{%
  \@ifdefinable{#1}{%
    \ZREF@Robust{#2}#1#3%
  }%
}
```

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

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

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

```
\newif\ifZREF@found
```

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

```
\def\ZREF@patch#1{%
  \ltx@IfUndefined{#1}{%
    \ltx@gobble
  }{%
    \expandafter\let\csname ZREF@org@#1\expandafter\endcsname
    \csname #1\endcsname
    \ltx@firstofone
  }%
}
```

6.2.4 Check for ε-TEX

The use of ε-TEX should be standard nowadays for LATEX. We test for ε-TEX in order to use its features later.

```
\ltx@IfUndefined{eTeXversion}{%
  \PackageError\ZREF@name{%
    Missing support for eTeX; package is abandoned%
  }{%
    Use a TeX compiler that support eTeX and enable eTeX %
    in the format.%
  }%
  \endinput
}{}%
\RequirePackage{etexcmds}[2007/09/09]
\ifetex@unexpanded
\else
  \PackageError\ZREF@name{%
    Missing e-TeX's \string\unexpanded.\MessageBreak
    Add \string\RequirePackage\string{etexcmds\string} before %
    \string\documentclass%
  }{%
    Probably you are using some package (e.g. ConTeXt) that %
    redefines \string\unexpanded%
  }%
  \expandafter\endinput
\fi
```

6.2.5 Auxiliary file stuff

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

```
\RequirePackage{auxhook}
\AddLineBeginAux{%
  \string\providecommand\string\zref@newlabel[2]{}%
}
```

\ZREF@RefPrefix

```
\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.
- LATEX's check for multiple references.
- LATEX's check for changed references.

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

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.

```
\ZREF@Robust\def\zref@newlist#1{%
  \zref@iflistundefined{#1}{%
    \@ifdefinable{Z@L@#1}{%
     \global\expandafter\let\csname Z@L@#1\endcsname\ltx@empty
     \PackageInfo\ZREF@name{New property list: #1}%
    }%
  }{%
    \PackageError\ZREF@name{%
      Property list `#1' already exists%
    }\@ehc
  }%
}
```

`\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.

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

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

```
\ZREF@Robust\def\zref@listexists#1{%
  \zref@iflistundefined{#1}{%
    \PackageError\ZREF@name{%
      Property list `#1' does not exist%
    }\@ehc
  }%
}
```

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

```
\ZREF@Robust\def\zref@iflistcontainsprop#1#2{%
  \zref@iflistundefined{#1}{%
    \ltx@secondoftwo
  }{%
    \begingroup\expandafter\endgroup
    \expandafter\in@
    \csname#2\expandafter\expandafter\expandafter\endcsname
    \expandafter\expandafter\expandafter{\csname Z@L@#1\endcsname}%
    \csname ltx@\ifin@ first\else second\fi oftwo\endcsname
  }%
}
```

`\zref@listforloop`

```
\def\zref@listforloop#1#2{%
  \zref@listexists{#1}{%
    \expandafter\expandafter\expandafter\@tfor
    \expandafter\expandafter\expandafter\zref@prop
    \expandafter\expandafter\expandafter:%
    \expandafter\expandafter\expandafter=%
    \csname Z@L@#1\endcsname
    \do{%
```

```
      \begingroup
        \escapechar=-1 %
        \edef\x{\endgroup
          \def\noexpand\zref@prop{%
            \expandafter\string\zref@prop
          }%
        }%
      \x
      #2\zref@prop
    }%
  }%
}
```

`\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.

```
\ZREF@Robust\def\zref@addprops#1#2{%
  \zref@listexists{#1}{%
    \comma@parse{#2}{%
      \zref@propexists\comma@entry{%
        \zref@iflistcontainsprop{#1}\comma@entry{%
          \PackageWarning\ZREF@name{%
            Property `\comma@entry' is already in list `#1'%
          }%
        }{%
          \begingroup\expandafter\endgroup
          \expandafter\g@addto@macro
          \csname Z@L@#1\expandafter\endcsname
          \expandafter{\csname\comma@entry\endcsname}%
        }%
      }%
      \ltx@gobble
    }%
  }%
}
```

`\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.

```
\ZREF@Robust\def\zref@addprop#1#2{%
  \zref@listexists{#1}{%
    \zref@propexists{#2}{%
      \zref@iflistcontainsprop{#1}{#2}{%
        \PackageWarning\ZREF@name{%
          Property `#2' is already in list `#1'%
        }%
      }{%
        \begingroup\expandafter\endgroup
        \expandafter\g@addto@macro
        \csname Z@L@#1\expandafter\endcsname
        \expandafter{\csname#2\endcsname}%
      }%
    }%
  }%
}
```

`\zref@localaddprops`

```
\ZREF@Robust\def\zref@localaddprops#1#2{%
  \zref@listexists{#1}{%
    \comma@parse{#2}{%
      \zref@propexists\comma@entry{%
        \zref@iflistcontainsprop{#1}\comma@entry{%
          \PackageWarning\ZREF@name{%
            Property `\comma@entry' is already in list `#1'%
```

```
          }%
        }{%
          \begingroup\expandafter\endgroup
          \expandafter\ltx@LocalAppendToMacro
          \csname Z@L@#1\expandafter\endcsname
          \expandafter{\csname\comma@entry\endcsname}%
        }%
      }%
      \ltx@gobble
    }%
  }%
}
```

`\zref@localaddprop`

```
\ZREF@Robust\def\zref@localaddprop#1#2{%
  \zref@listexists{#1}{%
    \zref@propexists{#2}{%
      \zref@iflistcontainsprop{#1}{#2}{%
        \PackageWarning\ZREF@name{%
          Property `#2' is already in list `#1'%
        }%
      }{%
        \begingroup\expandafter\endgroup
        \expandafter\ltx@LocalAppendToMacro
        \csname Z@L@#1\expandafter\endcsname
        \expandafter{\csname#2\endcsname}%
      }%
    }%
  }%
}
\ltx@IfUndefined{pdf@strcmp}{%
```

`\zref@delprop`

```
  \ZREF@Robust\def\zref@delprop{%
    \ZREF@delprop\gdef
  }%
```

`\zref@localdelprop`

```
  \ZREF@Robust\def\zref@localdelprop{%
    \ZREF@delprop\def
  }%
```

`\ZREF@delprop`

```
  \def\ZREF@delprop#1#2#3{%
    \zref@listexists{#2}{%
      \begingroup
        \escapechar=-1 %
        \def\ZREF@param{#3}%
        \@onelevel@sanitize\ZREF@param
        \toks@{}%
        \expandafter\expandafter\expandafter\ZREF@@delprop
        \csname Z@L@#2\endcsname!%
      \expandafter\endgroup
      \expandafter#1\csname Z@L@#2\expandafter\endcsname
      \expandafter{%
        \the\toks@
      }%
    }%
  }%
```

`\ZREF@@delprop`

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

`\ZREF@@@delprop`

```
  \def\ZREF@@@delprop#1#2{%
    \ifx#2!%
    \else
      \def\ZREF@temp{#1}%
      \@onelevel@sanitize\ZREF@temp
      \ifx\ZREF@param\ZREF@temp
      \else
        \toks@\expandafter{%
          \the\expandafter\toks@\csname#1\endcsname
        }%
      \fi
      \expandafter\ZREF@@delprop
    \fi
  }%
}{%
```

`\zref@delprop`

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

`\zref@localdelprop`

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

`\ZREF@delprop`

```
  \def\ZREF@delprop#1#2#3{%
    \zref@listexists{#2}{%
      \def\ZREF@param{#3}%
      \edef\ZREF@SavedEscapechar{\the\escapechar}%
      \escapechar=-1 %
      \expandafter#1\csname Z@L@#2%
      \expandafter\expandafter\expandafter\endcsname{%
        \expandafter\expandafter\expandafter\ZREF@@delprop
        \csname Z@L@#2\endcsname!%
      }%
      \escapechar=\ZREF@SavedEscapechar\relax
    }%
  }%
```

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

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

`\ZREF@@@delprop`

```
  \def\ZREF@@@delprop#1#2{%
    \ifx#2!%
    \else
      \ifnum\pdf@strcmp{#1}{\ZREF@param}=\ltx@zero
      \else
        \expandafter\noexpand\csname#1\endcsname
      \fi
      \expandafter\ZREF@@delprop
    \fi
  }%
```

```
}
```

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.

```
\def\zref@ifpropundefined#1{%
  \ltx@ifundefined{Z@E@#1}%
}
```

`\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.

```
\ZREF@Robust\def\zref@propexists#1{%
  \zref@ifpropundefined{#1}{%
    \PackageError\ZREF@name{%
      Property `#1' does not exist%
    }\@ehc
  }%
}
```

`\zref@newprop` A new property is declared by `\zref@newprop`, the property name ⟨*propname*⟩ 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.

```
\ZREF@Robust\def\zref@newprop{%
  \@ifstar{%
    \let\ZREF@X\noexpand
    \ZREF@newprop
  }{%
    \let\ZREF@X\ltx@empty
    \ZREF@newprop
  }%
}
```

`\ZREF@newprop`

```
\def\ZREF@newprop#1{%
  \edef\ZREF@P{#1}%
  \@onelevel@sanitize\ZREF@P
  \begingroup
  \ifx\ZREF@P\ZREF@par
    \@PackageError\ZREF@name{%
      Invalid property name `\ZREF@P'%
    }{%
      The property name `par' is not allowed %
      because of internal reasons.%
      \MessageBreak
      \@ehc
    }%
    \def\ZREF@@newprop[##1]##2{\endgroup}%
  \else
    \zref@ifpropundefined\ZREF@P{%
      \endgroup
      \PackageInfo\ZREF@name{%
```

```
        New property: \ZREF@P
      }%
    }{%
      \@PackageError\ZREF@name{%
        Property `\ZREF@P' already exists%
      }\@ehc
      \def\ZREF@@newprop[##1]##2{\endgroup}%
    }%
  \fi
  \@ifnextchar[\ZREF@@newprop{\ZREF@@newprop[\zref@default]}%
}
```

\ZREF@par

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

\ZREF@@newprop

```
\def\ZREF@@newprop[#1]{%
  \global\@namedef{Z@D@\ZREF@P}{#1}%
  \global\expandafter\let\csname Z@X@\ZREF@P\endcsname\ZREF@X
  \begingroup\expandafter\endgroup
  \expandafter\ZREF@@@newprop\csname\ZREF@P\endcsname
  \expandafter\gdef\csname Z@C@\ZREF@P\endcsname{}%
  \zref@setcurrent\ZREF@P
}
\def\ZREF@@@newprop#1{%
  \expandafter
  \gdef\csname Z@E@\ZREF@P\endcsname##1#1##2##3\ZREF@nil{##2}%
}
```

\zref@showprop

```
\ZREF@Robust\def\zref@showprop#1{%
  \zref@ifpropundefined{#1}{%
    \@PackageInfoNoLine{\ZREF@name}{%
      Show property `#1': <undefined>%
    }%
  }{%
    \begingroup
      \toks@\expandafter\expandafter\expandafter{%
        \csname Z@C@#1\endcsname
      }%
      \edef\ZREF@value{\the\toks@}%
      \ltx@onelevel@sanitize\ZREF@value
      \toks@\expandafter\expandafter\expandafter{%
        \csname Z@D@#1\endcsname
      }%
      \edef\ZREF@default{\the\toks@}%
      \ltx@onelevel@sanitize\ZREF@default
      \@PackageInfoNoLine{\ZREF@name}{%
        Show property `#1':\MessageBreak
        \expandafter\ifx\csname Z@X@#1\endcsname\ltx@empty
          Immediate %
        \else
          Delayed %
        \fi
        value: [\ZREF@value]\MessageBreak
        Default: [\ZREF@default]%
      }%
    \endgroup
  }%
}
```

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

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

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

```
\def\ZREF@getcurrent#1{%
  \romannumeral0%
  \ltx@ifundefined{Z@C@#1}{%
    \ltx@space
  }{%
    \expandafter\expandafter\expandafter\ltx@space
    \csname Z@C@#1\endcsname
  }%
}
```

\ZREF@u@getcurrent

```
\def\ZREF@wu@getcurrent#1{%
  \etex@unexpanded\expandafter\expandafter\expandafter{%
    \ZREF@getcurrent{#1}%
  }%
}
```

\zref@getcurrent

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

6.2.8 Reference generation

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

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

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

```
\ZREF@Robust\def\zref@labelbylist#1#2{%
  \@bsphack
    \zref@listexists{#2}{%
      \expandafter\expandafter\expandafter\ZREF@label
      \expandafter\expandafter\expandafter{%
        \csname Z@L@#2\endcsname
      }{#1}%
    }%
  \@esphack
}
```

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

```
\ZREF@Robust\def\zref@labelbyprops#1#2{%
  \@bsphack
    \begingroup
      \toks@{}%
      \comma@parse{#2}{%
        \zref@ifpropundefined\comma@entry{%
          \PackageWarning\ZREF@name{%
            Property `\comma@entry' is not known%
          }%
        }{%
          \toks@\expandafter{%
            \the\expandafter\toks@\csname\comma@entry\endcsname
          }%
```

```
        }%
        \ltx@gobble
      }%
    \expandafter\endgroup
    \expandafter\ZREF@label\expandafter{\the\toks@}{#1}%
  \@esphack
}
```

\zref@labelbykv

```
\ZREF@Robust\def\zref@labelbykv#1#2{%
  \@bsphack
    \begingroup
      \let\Z@L@ZREF@temp\ltx@empty
      \kvsetkeys{ZREF@LABEL}{#1}%
      \ifZREF@immediate
        \expandafter\zref@wrapper@immediate\expandafter{%
          \expandafter\ZREF@label\expandafter{\Z@L@ZREF@temp}{#2}%
        }%
      \else
        \expandafter\ZREF@label\expandafter{\Z@L@ZREF@temp}{#2}%
      \fi
    \endgroup
  \@esphack
}

\kv@define@key{ZREF@LABEL}{prop}{%
  \edef\ZREF@param{#1}%
  \zref@propexists\ZREF@param{%
    \zref@iflistcontainsprop{ZREF@temp}\ZREF@param{}{%
      \begingroup\expandafter\endgroup
      \expandafter\ltx@LocalAppendToMacro
      \expandafter\Z@L@ZREF@temp
      \expandafter{\csname\ZREF@param\endcsname}%
    }%
  }%
}
\kv@define@key{ZREF@LABEL}{list}{%
  \zref@listforloop{#1}{%
    \zref@iflistcontainsprop{ZREF@temp}\zref@prop{}{%
      \begingroup\expandafter\endgroup
      \expandafter\ltx@LocalAppendToMacro
      \expandafter\Z@L@ZREF@temp
      \expandafter{\csname\zref@prop\endcsname}%
    }%
    \ltx@gobble
  }%
}
\kv@define@key{ZREF@LABEL}{delprop}{%
  \zref@propexists{#1}{%
    \zref@localdelprop{ZREF@temp}{#1}%
  }%
}
\kv@define@key{ZREF@LABEL}{immediate}[true]{%
  \edef\ZREF@param{#1}%
  \ifx\ZREF@param\ZREF@true
    \ZREF@immediatetrue
  \else
    \ifx\ZREF@param\ZREF@false
      \ZREF@immediatefalse
    \else
      \PackageWarning\ZREF@name{%
        Option `immediate' expects `true' or `false'.\MessageBreak
        Ignoring invalid value `\ZREF@param'%
```

```
      }%
    \fi
  \fi
}
```

\ZREF@false

```
\def\ZREF@false{false}
```

\ZREF@true

```
\def\ZREF@true{true}
```

```
\kv@define@key{ZREF@LABEL}{values}[]{%
  \kv@parse{#1}{%
    \ifx\kv@value\relax
      \@PackageWarning\ZREF@name{%
        Missing value for property `\kv@key'%
      }%
      \expandafter\ltx@gobbletwo
    \else
      \expandafter\zref@setcurrent
    \fi
  }%
}
```

\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.

```
\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.

```
\ZREF@Robust{\long\def}\zref@wrapper@immediate#1{%
  \begingroup
    \ZREF@immediatetrue
    \let\ZREF@org@write\write
    \def\write{\immediate\ZREF@org@write}%
    #1%
  \endgroup
}
```

\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 34is made expandable in this case.

```
\def\ZREF@label#1#2{%
  \if@filesw
    \begingroup
      \ifZREF@immediate
        \let\ZREF@org@thepage\thepage
      \fi
      \protected@write\@auxout{%
        \ifZREF@immediate
          \let\thepage\ZREF@org@thepage
        \fi
        \let\ZREF@temp\ltx@empty
        \@tfor\ZREF@P:=#1\do{%
          \begingroup
            \escapechar=-1 %
            \edef\x{\endgroup
              \def\noexpand\ZREF@P{%
                \expandafter\string\ZREF@P
```

```
              }%
            }%
          \x
          \expandafter\ifx
              \csname
                \ifZREF@immediate
                  relax%
                \else
                  Z@X@\ZREF@P%
                \fi
              \endcsname
              \noexpand
            \expandafter\let\csname Z@C@\ZREF@P\endcsname\relax
          \fi
          \toks@\expandafter{\ZREF@temp}%
          \edef\ZREF@temp{%
            \the\toks@
            \ltx@backslashchar\ZREF@P{%
              \expandafter\noexpand\csname Z@C@\ZREF@P\endcsname
            }%
          }%
        }%
      }{%
        \string\zref@newlabel{#2}{\ZREF@temp}%
      }%
    \endgroup
  \fi
}
\def\ZREF@addtoks#1{%
  \toks@\expandafter\expandafter\expandafter{%
    \expandafter\the\expandafter\toks@#1%
  }%
}
```

6.2.9 Reference querying and extracting

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

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

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

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

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

```
\ZREF@IfDefinable\zifrefundefined\def{%
  #1{%
    \zref@wrapper@babel\ZREF@ifrefundefined{#1}%
  }%
}
```

\ZREF@ifrefundefined

```
\def\ZREF@ifrefundefined#1{%
  \zref@refused{#1}%
  \zref@ifrefundefined{#1}%
}
```

\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.

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

\ZREF@refused

```
\def\ZREF@refused#1{%
  \zref@ifrefundefined{#1}{%
    \protect\G@refundefinedtrue
    \@latex@warning{%
      Reference `#1' on page \thepage \space undefined%
    }%
  }{}%
}
```

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

```
\def\zref@ifrefcontainsprop#1#2{%
  \zref@ifrefundefined{#1}{%
    \ltx@secondoftwo
  }{%
    \expandafter\ZREF@ifrefcontainsprop
    \csname Z@E@#2\expandafter\endcsname
    \csname#2\expandafter\expandafter\expandafter\endcsname
    \expandafter\expandafter\expandafter{%
      \csname Z@R@#1\endcsname
    }%
  }%
}
\def\ZREF@ifrefcontainsprop#1#2#3{%
  \expandafter\ifx\expandafter\ZREF@novalue
  #1#3#2\ZREF@novalue\ZREF@nil\ltx@empty
    \expandafter\ltx@secondoftwo
  \else
    \expandafter\ltx@firstoftwo
  \fi
}
\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.

```
\def\ZREF@extract#1#2{%
  \romannumeral0%
  \ltx@ifundefined{Z@D@#2}{%
    \expandafter\ltx@space\zref@default
  }{%
    \expandafter\expandafter\expandafter\ZREF@@extract
    \expandafter\expandafter\expandafter{%
      \csname Z@D@#2\endcsname
    }{#1}{#2}%
  }%
}
```

\ZREF@@extract

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

\ZREF@wu@extract

```
\def\ZREF@wu@extract#1#2{%
  \etex@unexpanded\expandafter\expandafter\expandafter{%
    \ZREF@extract{#1}{#2}%
  }%
}
```

\zref@extract

```
\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.

```
\def\ZREF@extractdefault#1#2#3{%
  \romannumeral0%
  \zref@ifrefundefined{#1}\ltx@firstoftwo{%
    \zref@ifpropundefined{#2}\ltx@firstoftwo\ltx@secondoftwo
  }{%
    \ltx@space
    #3%
  }{%
    \expandafter\expandafter\expandafter\ltx@space
    \csname Z@E@#2\expandafter\expandafter\expandafter\endcsname
    \csname Z@R@#1\expandafter\endcsname
    \csname#2\endcsname{#3}\ZREF@nil
  }%
}
```

\ZREF@wu@extractdefault

```
\def\ZREF@wu@extractdefault#1#2#3{%
  \etex@unexpanded\expandafter\expandafter\expandafter{%
    \ZREF@extractdefault{#1}{#2}{#3}%
  }%
}
```

\zref@extractdefault

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

\zref@def@extract

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

\ZREF@def@extract

```
\def\ZREF@def@extract#1#2#3{%
  \zref@refused{#2}%
  \expandafter\expandafter\expandafter\def
  \expandafter\expandafter\expandafter#1%
  \expandafter\expandafter\expandafter{%
    \zref@extract{#2}{#3}%
  }%
}
```

\zref@def@extractdefault

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

\ZREF@def@extractdefault

```
\def\ZREF@def@extractdefault#1#2#3#4{%
  \zref@refused{#2}%
  \expandafter\expandafter\expandafter\def
  \expandafter\expandafter\expandafter#1%
  \expandafter\expandafter\expandafter{%
    \zref@extractdefault{#2}{#3}{#4}%
  }%
}
```

\ZREF@wrapper@unexpanded

```
\ZREF@Robust{\long\def}\ZREF@wrapper@unexpanded#1{%
  \let\zref@wrapper@unexpanded\ltx@firstofone
  \let\zref@getcurrent\ZREF@wu@getcurrent
  \let\zref@extractdefault\ZREF@wu@extractdefault
  \let\zref@extract\ZREF@wu@extract
  #1%
  \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded
  \let\zref@getcurrent\ZREF@getcurrent
  \let\zref@extractdefault\ZREF@extractdefault
  \let\zref@extract\ZREF@extract
}
```

\zref@wrapper@unexpanded

```
\ltx@IfUndefined{etex@unexpanded}{%
  \let\zref@wrapper@unexpanded\ltx@firstofone
}{%
  \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded
}
```

6.2.10 Compatibility with babel

\zref@wrapper@babel

```
\ZREF@Robust{\long\def}\zref@wrapper@babel#1#2{%
  \ifcsname if@safe@actives\endcsname
    \expandafter\ltx@firstofone
  \else
    \expandafter\ltx@secondoftwo
  \fi
  {%
    \if@safe@actives
      \expandafter\ltx@secondoftwo
    \else
      \expandafter\ltx@firstoftwo
    \fi
    {%
      \begingroup
        \csname @safe@activestrue\endcsname
        \edef\x{#2}%
      \expandafter\endgroup
      \expandafter\ZREF@wrapper@babel\expandafter{\x}{#1}%
    }%
  }{%
    #1{#2}%
  }%
}
\long\def\ZREF@wrapper@babel#1#2{%
  #2{#1}%
}
```

6.2.11 Unique counter support

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

```
\ZREF@Robust\def\zref@require@unique{%
  \@ifundefined{c@zref@unique}{%
      \newcounter{zref@unique}%
```

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

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

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

```
   \IfFormatAtLeastTF{2020/10/01}
    {%
     \AddToHook
       {include/after}
       {\setcounter{zref@unique}
         {\csname fp_to_int:n\endcsname{ceil(\c@zref@unique +100,-2)}}}
    }{}%
  }{}%
}
```

6.2.12 Utilities

\ZREF@number

```
\ltx@IfUndefined{numexpr}{%
  \def\ZREF@number#1{\number#1}%
}{%
  \def\ZREF@number#1{\the\numexpr(#1)\relax}%
}
```

6.2.13 Setup

\zref@setdefault Standard LaTeX 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`.

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

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

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

Main property list.

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

```
\ZREF@Robust\def\zref@setmainlist#1{%
  \def\ZREF@mainlist{#1}%
}
\zref@setmainlist{main}
```

Now we create the list.

```
\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 LaTeX 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

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

Properties

`\ZREF@NewPropAnchor`

```
\def\ZREF@NewPropAnchor{%
  \zref@newprop{anchor}{%
    \ltx@ifundefined{@currentHref}{}{\@currentHref}%
  }%
  \global\let\ZREF@NewPropAnchor\relax
}
```

`\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`

```
\def\ZREF@NewPropTitle{%
  \gdef\zref@titleref@current{}%
  \zref@newprop{title}{\zref@titleref@current}%
  \global\let\ZREF@NewPropTitle\relax
}
```

`\ZREF@NewPropTheotype`

```
\def\ZREF@NewPropTheotype{%
  \zref@newprop{theotype}{}%
  \global\let\ZREF@NewPropTheotype\relax
}
```

`\ZREF@NewPropPageValue`

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

Mark successful loading

```
\let\ZREF@base@ok=Y
```
⟨/base⟩

6.3 Module user

```
⟨*user⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-user}%
  [2025-06-08 v2.36 Module user for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
```

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

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

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

```
\newcommand*\zlabel{%
  \ifx\label\ltx@gobble
    \expandafter\ltx@gobble
  \else
    \expandafter\zref@wrapper@babel\expandafter\zref@label
  \fi
}%
```

`\zkvlabel`

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

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

```
\newcommand*{\zref}[2][default]{% robust because of optional argument
  \zref@propexists{#1}{%
    \zref@wrapper@babel\ZREF@zref{#2}{#1}%
  }%
}%
\def\ZREF@zref#1{%
  \zref@refused{#1}%
  \zref@extract{#1}%
}%
```

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

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

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

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

⟨/user⟩

6.4 Module abspage

```
⟨*abspage⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-abspage}%
  [2025-06-08 v2.36 Module abspage for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
```

Module `abspage` adds a new property `abspage` to the `main` property list for absolute page numbers. These are recorded by the help of the standard LaTeX shipout hooks. The count used as property is the standard LaTeX counter

\ReadonlyShipoutCounter, but we still provide also the counter abspage for compatibility with older code. The counter abspage must not go in the clear list of @ckpt that is used to set counters in .aux files of included TEX files.

```
\begingroup
 \@ifundefined{c@abspage}
  {%
   \let\@addtoreset\ltx@gobbletwo
   \newcounter{abspage}%
  }{}
\endgroup
\setcounter{abspage}{0}%
\AddToHook{shipout/before}{%
  \stepcounter{abspage}%
}%
\zref@newprop*{abspage}[0]{\the\ReadonlyShipoutCounter}%
\zref@addprop\ZREF@mainlist{abspage}%
```

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

⟨/abspage⟩

6.5 Module **counter**

⟨*counter⟩

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-counter}%
  [2025-06-08 v2.36 Module counter for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
```

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

```
\@ifl@t@r\fmtversion{2020-10-01}
 {
  \zref@newprop{counter}{\@currentcounter}
  \zref@addprop\ZREF@mainlist{counter}
 }
 {
  \zref@newprop{counter}{}
  \zref@addprop\ZREF@mainlist{counter}
  \AtBeginDocument{%
    \ZREF@patch{refstepcounter}{%
      \def\refstepcounter#1{%
        \zref@setcurrent{counter}{#1}%
        \ZREF@org@refstepcounter{#1}%
      }%
    }%
  }
 }
```

⟨/counter⟩

6.6 Module **lastpage**

⟨*lastpage⟩

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-lastpage}%
  [2025-06-08 v2.36 Module lastpage for zref (HO)]%
```

```
\RequirePackage{zref-base}[2019/11/29]
\RequirePackage{zref-abspage}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
```

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

```
\zref@newlist{LastPage}
\@ifl@t@r\fmtversion{2025-06-01}{%
\AddToHook {enddocument/afterlastpage}{%
  \if@filesw
    \begingroup
      \toks@\expandafter\expandafter\expandafter{%
        \expandafter\Z@L@main
        \Z@L@LastPage
      }%
      \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
    \endgroup
  \fi
}}
{\AddToHook {enddocument/afterlastpage}{%
  \if@filesw
    \begingroup
      \advance\c@page\m@ne
      \toks@\expandafter\expandafter\expandafter{%
        \expandafter\Z@L@main
        \Z@L@LastPage
      }%
      \expandafter\zref@wrapper@immediate\expandafter{%
        \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
      }%
    \endgroup
  \fi
}}
```

\zref@iflastpage

```
\def\zref@iflastpage#1{%
  \ifnum\zref@extractdefault{#1}{abspage}{-1}=%
        \zref@extractdefault{LastPage}{abspage}{-2} %
    \expandafter\ltx@firstoftwo
  \else
    \expandafter\ltx@secondoftwo
  \fi
}
```

\ziflastpage

```
\ZREF@IfDefinable\ziflastpage\def{%
  {\zref@wrapper@babel\ZREF@iflastpage}%
}
```

\ZREF@iflastpage

```
\def\ZREF@iflastpage#1{%
  \zref@refused{LastPage}%
  \zref@refused{#1}%
  \zref@iflastpage{#1}%
}
```

```
⟨/lastpage⟩
```

6.7 Module **thepage**

```
⟨*thepage⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-thepage}%
  [2025-06-08 v2.36 Module thepage for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\RequirePackage{atbegshi}[2011/10/05]
\RequirePackage{zref-abspage}[2019/11/29]
\zref@newlist{thepage}
\zref@addprop{thepage}{page}
\ZREF@NewPropPageValue
```

`\zref@thepage@atbegshi@hook`

```
\let\zref@thepage@atbegshi@hook\ltx@empty
\zref@addprop{thepage}{pagevalue}
\AddToHook{shipout/before}{%
  \AtBeginShipoutAddToBox{%
    \zref@thepage@atbegshi@hook
    \zref@labelbylist{thepage\the\value{abspage}}{thepage}%
  }%
}
```

`\zref@thepage@name`

```
\ltx@IfUndefined{numexpr}{%
  \def\zref@thepage@name#1{thepage\number#1}%
}{%
  \def\zref@thepage@name#1{thepage\the\numexpr#1}%
}
```

`\zref@thepage`

```
\def\zref@thepage#1{%
  \zref@extract{\zref@thepage@name{#1}}{page}%
}%
```

`\zref@thepage@refused`

```
\ZREF@Robust\def\zref@thepage@refused#1{%
  \zref@refused{\zref@thepage@name{#1}}%
}%
```

`\zthepage`

```
\ZREF@IfDefinable\zthepage\def{%
  #1{%
    \zref@thepage@refused{#1}%
    \zref@thepage{#1}%
  }%
}
⟨/thepage⟩
```

6.8 Module **nextpage**

```
⟨*nextpage⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-nextpage}%
  [2025-06-08 v2.36 Module nextpage for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
```

```
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\RequirePackage{zref-abspage}[2019/11/29]
\RequirePackage{zref-thepage}[2019/11/29]
\RequirePackage{zref-lastpage}[2019/11/29]
\RequirePackage{uniquecounter}[2009/12/18]
\UniqueCounterNew{znextpage}

\newcommand*{\znextpagesetup}{%
  \afterassignment\ZREF@np@setup@i
  \def\ZREF@np@call@unknown##1%
}
\def\ZREF@np@setup@i{%
  \afterassignment\ZREF@np@setup@ii
  \def\ZREF@np@call@nonext##1%
}
\def\ZREF@np@setup@ii{%
  \def\ZREF@np@call@next##1%
}
\def\ZREF@np@call@unknown#1{#1}
\def\ZREF@np@call@nonext#1{#1}
\def\ZREF@np@call@next#1{#1}
\ZREF@IfDefinable\znextpage\def{%
  {\UniqueCounterCall{znextpage}{\ZREF@nextpage}}%
}%
\newcommand*{\znonextpagename}{}
\newcommand*{\zunknownnextpagename}{\Z@D@page}
\def\ZREF@nextpage#1{%
  \begingroup
    \def\ZREF@refname@this{zref@np#1}%
    \zref@labelbyprops\ZREF@refname@this{abspage}%
    \chardef\ZREF@call=0 % unknown
    \ZREF@ifrefundefined\ZREF@refname@this{%
    }{%
      \edef\ZREF@pagenum@this{%
        \zref@extractdefault\ZREF@refname@this{abspage}{0}%
      }%
      \edef\ZREF@refname@next{%
        \zref@thepage@name{%
          \the\numexpr\ZREF@pagenum@this+1%
        }%
      }%
      \ifnum\ZREF@pagenum@this>0 %
        \ZREF@ifrefundefined{LastPage}{%
          \zref@ifrefundefined\ZREF@refname@next{%
          }{%
            \chardef\ZREF@call=2 % next page
          }%
        }{%
          \edef\ZREF@pagenum@last{%
            \zref@extractdefault{LastPage}{abspage}{0}%
          }%
          \ifnum\ZREF@pagenum@this<\ZREF@pagenum@last\ltx@space
            \ZREF@ifrefundefined\ZREF@refname@next{%
            }{%
              \chardef\ZREF@call=2 % next page
            }%
          \else
            \ifnum\ZREF@pagenum@this=\ZREF@pagenum@this\ltx@space
              \chardef\ZREF@call=1 % no next page
```

```
            \fi
          \fi
        }%
      \fi
    }%
    \edef\x{%
      \endgroup
      \ifcase\ZREF@call
        \noexpand\ZREF@np@call@unknown{%
          \noexpand\zunknownnextpagename
        }%
      \or
        \noexpand\ZREF@np@call@nonext{%
          \noexpand\znonextpagename
        }%
      \else
        \noexpand\ZREF@np@call@next{%
          \noexpand\zref@extract{\ZREF@refname@next}{page}%
        }%
      \fi
    }%
  \x
}
```

⟨/nextpage⟩

6.9 Module **totpages**

```
⟨*totpages⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-totpages}%
  [2025-06-08 v2.36 Module totpages for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
```

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

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

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

```
\newcommand*{\ztotpages}{%
  \zref@extractdefault{LastPage}{abspage}{0}%
}
```

Also we mark the reference `LastPage` as used:

```
\AtBeginDocument{%
  \zref@refused{LastPage}%
}
```

⟨/totpages⟩

6.10 Module **pagelayout**

```
⟨*pagelayout⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-pagelayout}%
  [2025-06-08 v2.36 Module pagelayout for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
```

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

6.10.1 Define layout properties

```
\def\ZREF@temp#1{%
  \begingroup
    \escapechar=-1 %
  \ltx@ifundefined{\string#1}{\endgroup}{%
    \edef\x{%
      \endgroup
      \noexpand\zref@newprop*{\string#1}%
                               [\noexpand\number\noexpand#1]% hash-ok
                               {\noexpand\number\noexpand#1}%
      \noexpand\zref@addprop{thepage}{\string#1}%
    }%
    \x
  }%
}
\ZREF@temp\mag
\ZREF@temp\paperwidth
\ZREF@temp\paperheight
\ZREF@temp\stockwidth % memoir.cls, crop.sty
\ZREF@temp\stockheight % memoir.cls, crop.sty
\ZREF@temp\mediawidth % VTeX
\ZREF@temp\mediaheight % VTeX
\ifluatex
\zref@newprop*{pdfvorigin}%
                               [\number\pdfvariable vorigin]% hash-ok
                               {\number\pdfvariable vorigin}%
\zref@addprop{thepage}{pdfvorigin}
\zref@newprop*{pdfhorigin}%
                               [\number\pdfvariable horigin]% hash-ok
                               {\number\pdfvariable horigin}%
\zref@addprop{thepage}{pdfhorigin}
\zref@newprop*{pdfpageheight}%
                               [\number\pageheight]% hash-ok
                               {\number\pageheight}%
\zref@addprop{thepage}{pdfpageheight}
\zref@newprop*{pdfpagewidth}%
                               [\number\pagewidth]% hash-ok
                               {\number\pagewidth}%
\zref@addprop{thepage}{pdfpagewidth}
\else
\ZREF@temp\pdfpagewidth
\ZREF@temp\pdfpageheight
\ZREF@temp\pdfhorigin
\ZREF@temp\pdfvorigin
\fi
\ZREF@temp\hoffset
\ZREF@temp\voffset
\ZREF@temp\topmargin
\ZREF@temp\oddsidemargin
\ZREF@temp\evensidemargin
\ZREF@temp\textwidth
\ZREF@temp\textheight
\ZREF@temp\headheight
\ZREF@temp\headsep
\ZREF@temp\footskip
\ZREF@temp\marginparwidth
\ZREF@temp\marginparsep
\ZREF@temp\columnwidth
```

```
\ZREF@temp\columnsep
\ZREF@temp\trimedge % memoir.cls
\ZREF@temp\spinemargin % memoir.cls
\ZREF@temp\foremargin % memoir.cls
\ZREF@temp\trimtop % memoir.cls
\ZREF@temp\uppermargin % memoir.cls
\ZREF@temp\headmargin % memoir.cls
\IfFormatAtLeastTF{2020/10/01}
 {
   \zref@newprop*{outputboxwd}[0pt]{\ShipoutBoxWidth}
   \zref@newprop*{outputboxht}[0pt]{\ShipoutBoxHeight}
   \zref@newprop*{outputboxdp}[0pt]{\ShipoutBoxDepth}
 }
 {
   \zref@newprop*{outputboxwd}[0pt]{\AtBeginShipoutBoxWidth}
   \zref@newprop*{outputboxht}[0pt]{\AtBeginShipoutBoxHeight}
   \zref@newprop*{outputboxdp}[0pt]{\AtBeginShipoutBoxDepth}
 }
\zref@addprops{thepage}{outputboxwd,outputboxht,outputboxdp}
```

`\ifZREF@pl@list`

```
\ltx@newif\ifZREF@pl@list
```

`\zref@listpagelayout`

```
\ZREF@IfDefinable\zlistpagelayout\def{%
  {\global\ZREF@pl@listtrue}%
}
```

`\ZREF@pl@AfterLastShipout`

```
\def\ZREF@pl@AfterLastShipout{%
  \ifZREF@pl@list
    \edef\ZREF@page@max{\the\value{abspage}}%
    \ltx@ifundefined{ZREF@org@testdef}{%
      \let\ZREF@org@testdef\@testdef
      \def\@testdef##1##2##3{%
        \ZREF@org@testdef{##1}{##2}{##3}%
        \def\ZREF@temp{##1}%
        \ifx\ZREF@temp\ZREF@RefPrefix
          \expandafter\gdef\csname##1@##2\endcsname{##3}%
        \fi
      }%
    }{}%
    \AddToHook{enddocument/afteraux}{\ZREF@pl@AtVeryEnd}%
  \fi
}
```

`\ZREF@pl@AtVeryEnd`

```
\def\ZREF@pl@AtVeryEnd{%
  \begingroup
    \toks@{Page layout parameters:\MessageBreak}%
    \count@=1 %
    \ZREF@pl@ListPage
    \edef\x{\endgroup
      \noexpand\@PackageInfoNoLine{zref-pagelayout}{\the\toks@}%
    }%
  \x
}
```

`\ZREF@pl@ListPage`

```
\def\ZREF@pl@ListPage{%
  \edef\x{%
```

```
    \toks@={%
      \the\toks@
      Page \the\count@:\noexpand\MessageBreak
      \zref@ifrefundefined{thepage\the\count@}{}{%
        \ltx@space\ltx@space mag = %
        \zref@extract{thepage\the\count@}{mag}%
        \noexpand\MessageBreak
        \ZREF@pl@ListEntry{paperwidth}%
        \ZREF@pl@ListEntry{paperheight}%
        \ZREF@pl@ListEntry{stockwidth}%
        \ZREF@pl@ListEntry{stockheight}%
        \ZREF@pl@ListEntry{mediawidth}%
        \ZREF@pl@ListEntry{mediaheight}%
        \ZREF@pl@ListEntry{pdfpagewidth}%
        \ZREF@pl@ListEntry{pdfpageheight}%
        \ZREF@pl@ListEntry{pdfhorigin}%
        \ZREF@pl@ListEntry{pdfvorigin}%
        \ZREF@pl@ListEntry{hoffset}%
        \ZREF@pl@ListEntry{voffset}%
        \ZREF@pl@ListEntry{topmargin}%
        \ZREF@pl@ListEntry{oddsidemargin}%
        \ZREF@pl@ListEntry{evensidemargin}%
        \ZREF@pl@ListEntry{textwidth}%
        \ZREF@pl@ListEntry{textheight}%
        \ZREF@pl@ListEntry{headheight}%
        \ZREF@pl@ListEntry{headsep}%
        \ZREF@pl@ListEntry{footskip}%
        \ZREF@pl@ListEntry{marginparwidth}%
        \ZREF@pl@ListEntry{marginparsep}%
        \ZREF@pl@ListEntry{columnwidth}%
        \ZREF@pl@ListEntry{columnsep}%
        \ZREF@pl@ListEntry{trimedge}%
        \ZREF@pl@ListEntry{spinemargin}%
        \ZREF@pl@ListEntry{foremargin}%
        \ZREF@pl@ListEntry{trimtop}%
        \ZREF@pl@ListEntry{uppermargin}%
        \ZREF@pl@ListEntry{headmargin}%
      }%
    }%
  }\x
  \ifnum\ZREF@page@max>\count@
    \advance\count@ by\ltx@one
  \else
    \expandafter\ltx@gobble
  \fi
  \ZREF@pl@ListPage
}
```

\ZREF@pl@ListEntry

```
\def\ZREF@pl@ListEntry#1{%
  \zref@ifpropundefined{#1}{%
  }{%
    \zref@ifrefcontainsprop{thepage\the\count@}{#1}{%
      \ltx@space\ltx@space#1 = %
      \zref@extract{thepage\the\count@}{#1}sp = %
      \the\dimexpr\zref@extract{thepage\the\count@}{#1}sp\relax
      \noexpand\MessageBreak
    }{}%
  }%
}

\AddToHook {enddocument/afterlastpage}{%
  \ZREF@pl@AfterLastShipout
```

```
}
```
⟨/pagelayout⟩

6.11 Module **pageattr**

```
⟨*pageattr⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-pageattr}%
  [2025-06-08 v2.36 Module pageattr for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\RequirePackage{iftex}[2019/11/07]%
\let\ZREF@temp=N%
\ifluatex
\expandafter\@firstoftwo
\else
\expandafter\@secondoftwo
\fi
{%luatex
 \RequirePackage{zref-thepage}[2019/11/29]
 \RequirePackage{zref-lastpage}[2019/11/29]%
 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfvariable pageattr}}%
 \zref@addprop{thepage}{pdfpageattr}%
 \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfvariable pagesattr}}%
 \zref@addprop{LastPage}{pdfpagesattr}%
 \let\ZREF@temp=Y%
}
{%other
 \ltx@IfUndefined{pdfpageattr}{%
  \@PackageInfoNoLine{zref-pageattr}{%
    \string\pdfpageattr\space is not available%
  }%
  \def\zref@pdfpageattr#1{}%
  \def\zref@pdfpageattr@used#1{}%
 }{%
  \RequirePackage{zref-thepage}[2019/11/29]%
  \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfpageattr}}%
  \zref@addprop{thepage}{pdfpageattr}%
  \let\ZREF@temp=Y%
 }
 \ltx@IfUndefined{pdfpagesattr}{%
  \@PackageInfoNoLine{zref-pageattr}{%
    \string\pdfpagesattr\space is not available%
  }%
  \def\zref@pdfpagesattr{}%
  \def\zref@pdfpagesattr@used{}%
 }{%
  \RequirePackage{zref-lastpage}[2019/11/29]%
  \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfpagesattr}}%
  \zref@addprop{LastPage}{pdfpagesattr}%
  \let\ZREF@temp=Y%
 }%
}%
\ifx\ZREF@temp N%
  \expandafter\endinput
\fi
\RequirePackage{zref-abspage}[2019/11/29]
```

```
\RequirePackage{pdftexcmds}[2010/04/01]
\let\ZREF@temp=Y%
\ltx@IfUndefined{pdf@escapehex}{\let\ZREF@temp=N}{}
\ltx@IfUndefined{pdf@unescapehex}{\let\ZREF@temp=N}{}
\ifx\ZREF@temp N%
  \let\zref@hex\ltx@firstofone
  \let\zref@unhex\ltx@firstofone
\else
  \let\zref@hex\pdf@escapehex
  \let\zref@unhex\pdf@unescapehex
\fi
```

`\ifZREF@pa@list`

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

`\zref@listpageattr`

```
\ZREF@IfDefinable\zlistpageattr\def{%
  {\ZREF@pa@listtrue}%
}
```

`\ZREF@pa@AfterLastShipout`

```
\def\ZREF@pa@AfterLastShipout{%
  \ifZREF@pa@list
    \edef\ZREF@page@max{\the\value{abspage}}%
    \ltx@ifundefined{ZREF@org@testdef}{%
      \let\ZREF@org@testdef\@testdef
      \def\@testdef##1##2##3{%
        \ZREF@org@testdef{##1}{##2}{##3}%
        \def\ZREF@temp{##1}%
        \ifx\ZREF@temp\ZREF@RefPrefix
          \expandafter\xdef\csname##1@##2\endcsname{##3}%
        \fi
      }%
    }{}%
    \AddToHook{enddocument/afteraux}{\ZREF@pa@AtVeryEnd}%
  \fi
}
```

`\ZREF@pa@AtVeryEnd`

```
\let\ZREF@temp=Y%
\ltx@IfUndefined{pdfpageattr}{}{\let\ZREF@temp=N}
\ifluatex \let\ZREF@temp=N \fi
\ifx\ZREF@temp Y
 \expandafter\@firstoftwo
\else
 \expandafter\@secondoftwo
\fi
 {%
  \def\ZREF@pa@AtVeryEnd{}%
 }
 {%
  \def\ZREF@pa@AtVeryEnd{%
    \begingroup
      \toks@{List of \ltx@backslashchar
             \ifluatex pdfvariable\else pdf\fi
             pdfpageattr:\MessageBreak}%
      \count@=1 %
      \ZREF@pa@ListPage
      \edef\x{\endgroup
        \noexpand\@PackageInfoNoLine{zref-pageattr}{%
          \the\toks@
        }%
      }%
```

```
      \x
    }%
```

\zref@pageattr

```
\def\zref@pdfpageattr#1{%
  \zref@unhex{%
    \zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}%
  }%
}
% compatibility, \zref@pageattr was defined in older versions
\let\zref@pageattr\zref@pdfpageattr
```

\zref@pageattr@used

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

\ZREF@pa@ListPage

```
  \def\ZREF@pa@ListPage{%
    \edef\x{%
      \toks@={%
        \the\toks@
        Page \the\count@:%
        \noexpand\MessageBreak
        \zref@ifrefundefined{thepage\the\count@}{}{%
          <<\zref@pdfpageattr\count@>>%
          \noexpand\MessageBreak
        }%
      }%
    }\x
    \ifnum\ZREF@page@max>\count@
      \advance\count@ by\ltx@one
    \else
      \expandafter\ltx@gobble
    \fi
    \ZREF@pa@ListPage
  }%
}

\let\ZREF@temp=Y%
\ltx@IfUndefined{pdfpagesattr}{}{\let\ZREF@temp=N}
\ifluatex \let\ZREF@temp=N \fi
\ifx\ZREF@temp N
 \expandafter\@firstofone
\fi
 {%
```

\zref@pdfpagesattr

```
  \def\zref@pdfpagesattr{%
    \zref@unhex{%
      \zref@extract{LastPage}{pdfpagesattr}%
    }%
  }%
```

\zref@pdfpagesattr@used

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

  \ltx@LocalAppendToMacro\ZREF@pa@AtVeryEnd{%
    \@PackageInfoNoLine{zref-pageattr}{%
      \ltx@backslashchar
```

```
      \ifluatex pdfvariable\else pdf\fi
      pagesattr:\MessageBreak
      <<\zref@pdfpagesattr>>%
      \MessageBreak
    }%
  }%
}
\AddToHook {enddocument/afterlastpage}{%
  \ZREF@pa@AfterLastShipout
}
```

⟨/pageattr⟩

6.12 Module **marks**

```
⟨*marks⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-marks}%
  [2025-06-08 v2.36 Module marks for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\newcommand*{\zref@marks@register}[3][]{%
  \edef\ZREF@TempName{#1}%
  \edef\ZREF@TempNum{\ZREF@number{#2}}%
  \ifnum\ZREF@TempNum<\ltx@zero %
    \PackageError\ZREF@name{%
      \string\zref@marks@register\ltx@space is called with invalid%
      \MessageBreak
      marks register number (\ZREF@TempNum)%
    }{%
      Use `0' or the command, defined by \string\newmarks.\MessageBreak
      \@ehc
    }%
  \else
    \ifx\ZREF@TempName\ltx@empty
      \edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}%
    \else
      \edef\ZREF@TempName{marks\ZREF@TempName}%
    \fi
    \ZREF@MARKS@DefineProp{top}%
    \ZREF@MARKS@DefineProp{first}%
    \ZREF@MARKS@DefineProp{bot}%
    \kv@parse{#3}{%
      \ifx\kv@value\relax
        \def\kv@value{top,first,bot}%
      \fi
      \edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}%
      \ifx\ZREF@temp\ZREF@STAR
        \edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}%
        \zref@newlist\kv@key
      \fi
      \expandafter\comma@parse\expandafter{\kv@value}{%
        \ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else
                \ifx\comma@entry\ZREF@NAME@first 1\else
                \ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space
          \PackageWarning{zref-marks}{%
            Use `top', `first' or `bot' for the list values%
            \MessageBreak
            in the third argument of \string\zref@marks@register.%
            \MessageBreak
```

```
            Ignoring unknown value `\comma@entry'%
          }%
        \else
          \zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}%
        \fi
        \ltx@gobble
      }%
      \ltx@gobbletwo
    }%
  \fi
}
\def\ZREF@STAR{*}
\def\ZREF@NAME@top{top}
\def\ZREF@NAME@first{first}
\def\ZREF@NAME@bot{bot}
\def\ZREF@MARKS@DefineProp#1{%
  \zref@ifpropundefined{#1\ZREF@TempName}{%
    \ifnum\ZREF@TempNum=\ltx@zero
      \begingroup
        \edef\x{\endgroup
          \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
            \expandafter\noexpand\csname#1mark\endcsname
          }%
        }%
      \x
    \else
      \begingroup
        \edef\x{\endgroup
          \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
            \expandafter\noexpand\csname#1marks\endcsname
            \ZREF@TempNum
          }%
        }%
      \x
    \fi
  }{%
    \PackageWarning{zref-marks}{%
      \string\zref@marks@register\ltx@space does not generate the%
      \MessageBreak
      new property `#1\ZREF@TempName', because\MessageBreak
      it is already defined%
    }%
  }%
}
```

⟨/marks⟩

6.13 Module **runs**

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

⟨*runs⟩

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-runs}%
  [2025-06-08 v2.36 Module runs for zref (HO)]%
```

`\zruns`

```
\providecommand*{\zruns}{0}%
\AtBeginDocument{%
  \edef\zruns{\number\numexpr\zruns+1}%
  \begingroup
    \def\on@line{}%
    \PackageInfo{zref-runs}{LaTeX runs: \zruns}%
```

```
    \if@filesw
      \immediate\write\@mainaux{%
        \string\gdef\string\zruns{\zruns}%
      }%
    \fi
  \endgroup
}
```

⟨/runs⟩

6.14 Module perpage

```
⟨*perpage⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-perpage}%
  [2025-06-08 v2.36 Module perpage for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\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.

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

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

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

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

```
\newcounter{zpage}
```

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

```
\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`.

```
\let\ZREF@org@stepcounter\stepcounter
\def\stepcounter#1{%
  \ifcsname @stepcounterhook@#1\endcsname
    \csname @stepcounterhook@#1\endcsname
  \fi
  \ZREF@org@stepcounter{#1}%
}
```

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

```
\IfFormatAtLeastTF{2020/10/01}
  {
    \AddToHook{package/amstext/after}
      {\def\stepcounter#1{%
        \iffirstchoice@
         \ifcsname @stepcounterhook@#1\endcsname
           \csname @stepcounterhook@#1\endcsname
         \fi
         \addtocounter{#1}\@ne
         \begingroup \let\@elt\@stpelt \csname cl@#1\endcsname \endgroup
       \fi
```

```
    }}
  }{}
%     \end{macrocode}
%
%    \cs{@stpelt} must be adapted due to the change in latex
%    2015-01, see https://github.com/ho-tex/zref/issues/26
%    \begin{macrocode}
\let\ZREF@org@@stpelt\@stpelt
\def\@stpelt#1{%
  \ifcsname ZREF@perpage@#1\endcsname
    \begingroup
      \let\stepcounter\ZREF@org@stepcounter
      \ZREF@org@@stpelt{#1}%
    \endgroup
    \expandafter\ltx@gobbletwo
  \fi
  \ZREF@org@@stpelt{#1}%
}
```

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

```
\ZREF@IfDefinable\zmakeperpage\def{%
  {%
    \@ifnextchar[\ZREF@makeperpage@opt{\ZREF@@makeperpage[\ltx@zero]}%
  }%
}
```

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

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

```
\def\ZREF@@makeperpage[#1]#2{%
  \@ifundefined{@stepcounterhook@#2}{%
    \expandafter\gdef\csname @stepcounterhook@#2\endcsname{}%
  }{}%
  \expandafter\gdef\csname ZREF@perpage@#2\endcsname{%
    \ZREF@@perpage@step{#2}{#1}%
  }%
  \expandafter\g@addto@macro\csname @stepcounterhook@#2\endcsname{%
    \ifcsname ZREF@perpage@#2\endcsname
      \csname ZREF@perpage@#2\endcsname
    \fi
  }%
}
```

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

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

First the reference is generated.

```
  \global\advance\c@zref@unique\ltx@one
  \begingroup
    \expandafter
    \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 kwown, we use the page counter as approximation.

```
    \zref@ifrefundefined\thezref@unique{%
      \global\c@zpage=\c@page
      \global\let\thezpage\thepage
      \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
        \number\c@abspage
      }%
    }{%
```

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

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

Page changes are detected by a changed absolute page number.

```
    \expandafter\ifx\csname ZREF@abspage@#1\expandafter\endcsname
                    \csname ZREF@currentabspage@#1\endcsname
    \else
      \global\csname c@#1\endcsname=#2\relax
      \global\expandafter\let
          \csname ZREF@currentabspage@#1\expandafter\endcsname
          \csname ZREF@abspage@#1\endcsname
    \fi
  \endgroup
}
```

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

```
\ZREF@IfDefinable\zunmakeperpage\def{%
  #1{%
    \global\expandafter
    \let\csname ZREF@perpage@#1\endcsname\@undefined
  }%
}
```

⟨/perpage⟩

6.15 Module **titleref**

⟨*titleref⟩

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-titleref}%
  [2025-06-08 v2.36 Module titleref for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\RequirePackage{gettitlestring}[2009/12/08]
```

6.15.1 Implementation

```
\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.

```
\ZREF@NewPropTitle
```

```
\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.

```
\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`.

```
\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.

```
\ZREF@Robust\def\zref@titleref@cleanup#1{%
  \begingroup
   \toks@\expandafter{%
     \ZREF@titleref@hook
     #1%
   }%
 \expandafter\endgroup
 \expandafter\def\expandafter\ZREF@titleref@hook\expandafter{%
   \the\toks@
 }%
}%
```

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

```
\newif\ifzref@titleref@stripperiod
\zref@titleref@stripperiodtrue
```

`\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.

```
\ZREF@Robust\def\zref@titleref@setcurrent#1{%
 \ifzref@titleref@expand
   \GetTitleStringExpand{#1}%
 \else
   \GetTitleStringNonExpand{#1}%
 \fi
 \edef\zref@titleref@current{%
   \detokenize\expandafter{\GetTitleStringResult}%
 }%
 \ifzref@titleref@stripperiod
   \edef\zref@titleref@current{%
     \expandafter\ZREF@stripperiod\zref@titleref@current
     \ltx@empty.\ltx@empty\@nil
   }%
 \fi
}%
\GetTitleStringDisableCommands{%
 \ZREF@titleref@hook
}
```

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

```
\def\ZREF@stripperiod#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.

```
\define@key{ZREF@TR}{expand}[true]{%
  \csname zref@titleref@expand#1\endcsname
}%
\define@key{ZREF@TR}{stripperiod}[true]{%
  \csname zref@titleref@stripperiod#1\endcsname
}%
\define@key{ZREF@TR}{cleanup}{%
  \zref@titleref@cleanup{#1}%
}%
\define@key{ZREF@TR}{title}{%
  \def\zref@titleref@current{#1}%
}%
\ZREF@IfDefinable\ztitlerefsetup\def{%
  {\kvsetkeys{ZREF@TR}}%
}%
```

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

```
\ZREF@IfDefinable\ztitleref\def{%
  {\zref@wrapper@babel\ZREF@titleref}%
}%
\def\ZREF@titleref#1{%
  \begingroup
    \zref@refused{#1}%
    \let\label\ltx@gobble
    \zref@extract{#1}{title}%
  \endgroup
}%
```

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.

```
\AtBeginDocument{%
  \ZREF@patch{@caption}{%
    \long\def\@caption#1[#2]{%
      \zref@titleref@setcurrent{#2}%
      \ZREF@org@@caption{#1}[{#2}]%
    }%
  }%
```

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

```
  \ZREF@patch{@part}{%
    \def\@part[#1]{%
      \zref@titleref@setcurrent{#1}%
      \ZREF@org@@part[{#1}]%
    }%
  }%
  \ZREF@patch{@chapter}{%
    \def\@chapter[#1]{%
      \zref@titleref@setcurrent{#1}%
```

```
      \ZREF@org@@chapter[{#1}]%
    }%
  }%
  \ZREF@patch{@sect}{%
    \def\@sect#1#2#3#4#5#6[#7]{%
      \zref@titleref@setcurrent{#7}%
      \ZREF@org@@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]%
    }%
  }%
```

The star versions of the section commands.

```
  \ZREF@patch{@spart}{%
    \def\@spart#1{%
      \zref@titleref@setcurrent{#1}%
      \ZREF@org@@spart{#1}%
    }%
  }%
  \ZREF@patch{@schapter}{%
    \def\@schapter#1{%
      \zref@titleref@setcurrent{#1}%
      \ZREF@org@@schapter{#1}%
    }%
  }%
  \ZREF@patch{@ssect}{%
    \def\@ssect#1#2#3#4#5{%
      \zref@titleref@setcurrent{#5}%
      \ZREF@org@@ssect{#1}{#2}{#3}{#4}{#5}%
    }%
  }%
```

6.15.4 Environment description

```
  \ZREF@patch{descriptionlabel}{%
    \def\descriptionlabel#1{%
      \zref@titleref@setcurrent{#1}%
      \ZREF@org@descriptionlabel{#1}%
    }%
  }%
```

6.15.5 Class memoir

```
  \@ifclassloaded{memoir}{%
    \ltx@IfUndefined{ifheadnameref}{}{%
      \def\@chapter[#1]#2{%
        \ltx@IfUndefined{ch@pt@c}{%
          \zref@titleref@setcurrent{#1}%
        }{%
          \ifx\ch@pt@c\ltx@empty
            \zref@titleref@setcurrent{#2}%
          \else
            \def\NR@temp{#1}%
            \ifx\NR@temp\ltx@empty
              \expandafter\zref@titleref@setcurrent
              \expandafter{\ch@pt@c}%
            \else
              \ifheadnameref
                \zref@titleref@setcurrent{#1}%
              \else
                \expandafter\zref@titleref@setcurrent
                \expandafter{\ch@pt@c}%
              \fi
            \fi
          \fi
        }%
```

```
      \ZREF@org@@chapter[{#1}]{#2}%
    }%
    \ZREF@patch{M@sect}{%
      \def\M@sect#1#2#3#4#5#6[#7][#8]{%
        \ifheadnameref
          \zref@titleref@setcurrent{#8}%
        \else
          \zref@titleref@setcurrent{#7}%
        \fi
        \ZREF@org@M@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}][{#8}]%
      }%
    }%
  }%
}{}%
```

6.15.6 Class beamer

```
\@ifclassloaded{beamer}{%
  \ZREF@patch{beamer@section}{%
    \long\def\beamer@section[#1]{%
      \zref@titleref@setcurrent{#1}%
      \ZREF@org@beamer@section[{#1}]%
    }%
  }%
  \ZREF@patch{beamer@subsection}{%
    \long\def\beamer@subsection[#1]{%
      \zref@titleref@setcurrent{#1}%
      \ZREF@org@beamer@subsection[{#1}]%
    }%
  }%
  \ZREF@patch{beamer@subsubsection}{%
    \long\def\beamer@subsubsection[#1]{%
      \zref@titleref@setcurrent{#1}%
      \ZREF@org@beamer@subsubsection[{#1}]%
    }%
  }%
}{}%
```

6.15.7 Package titlesec

```
\@ifpackageloaded{titlesec}{%
  \ZREF@patch{ttl@sect@i}{%
    \def\ttl@sect@i#1#2[#3]#4{%
      \zref@titleref@setcurrent{#4}%
      \ZREF@org@ttl@sect@i{#1}{#2}[{#3}]{#4}%
    }%
  }%
  \ZREF@patch{ttl@straight@i}{%
    \def\ttl@straight@i#1[#2]#3{%
      \def\ZREF@temp{#2}%
      \ifx\ZREF@temp\ltx@empty
        \zref@titleref@setcurrent{#3}%
      \else
        \zref@titleref@setcurrent{#2}%
      \fi
      \ZREF@org@ttl@straight@i{#1}[{#2}]{#3}%
    }%
  }%
}{}%
```

6.15.8 Package longtable

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

```
\@ifpackageloaded{longtable}{%
```

```
    \ZREF@patch{LT@c@ption}{%
      \def\LT@c@ption#1[#2]#3{%
        \ZREF@org@LT@c@ption{#1}[{#2}]{#3}%
        \zref@titleref@setcurrent{#2}%
      }%
    }%
  }{}%
```

6.15.9 Package listings

Package listings: support for its caption.

```
  \@ifpackageloaded{listings}{%
    \ZREF@patch{lst@MakeCaption}{%
      \def\lst@MakeCaption{%
        \ifx\lst@label\ltx@empty
        \else
          \expandafter\zref@titleref@setcurrent\expandafter{%
            \lst@@caption
          }%
        \fi
        \ZREF@org@lst@MakeCaption
      }%
    }%
  }{}%
```

6.15.10 Theorems

```
  \ZREF@patch{@opargbegintheorem}{%
    \def\@opargbegintheorem#1#2#3{%
      \zref@titleref@setcurrent{#3}%
      \ZREF@org@@opargbegintheorem{#1}{#2}{#3}%
    }%
  }%
  \@ifpackageloaded{amsthm}{%
    \begingroup
      \edef\x{macro:\string#1\string#2[\string#3]}%
      \@onelevel@sanitize\x
      \def\y#1->#2\@nil{#1}%
      \edef\z{\expandafter\y\meaning\@begintheorem->\@nil}%
      \@onelevel@sanitize\z
    \expandafter\endgroup
    \ifx\x\z
      \ZREF@patch{@begintheorem}{%
        \def\@begintheorem#1#2[#3]{%
          \zref@titleref@setcurrent{#3}%
          \ZREF@org@@begintheorem{#1}{#2}[{#3}]%
        }%
      }%
    \fi
  }{}%
}
```

⟨/titleref⟩

6.16 Module xr

```
⟨*xr⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-xr}%
  [2025-06-08 v2.36 Module xr for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
```

```
2082   \expandafter\endinput
2083 \fi
2084 \RequirePackage{keyval}
2085 \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.

```
2086 \zref@newprop{url}{}%
2087 \zref@newprop{urluse}{}%
2088 \zref@newprop{externaldocument}{}%
```

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

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

```
2089 \providecommand*{\zref@xr@ext}{%
2090   \ltx@ifundefined{XR@ext}{pdf}{\XR@ext}%
2091 }%
```

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

```
2092 \newif\ifZREF@xr@zreflabel

2093 \SetupKeyvalOptions{%
2094   family=ZREF@XR,%
2095   prefix=ZREF@xr@%
2096 }
2097 \DeclareBoolOption[true]{tozreflabel}
2098 \DeclareBoolOption[false]{toltxlabel}
2099 \DeclareBoolOption{verbose}
2100 \define@key{ZREF@XR}{ext}{%
2101   \def\zref@xr@{#1}%
2102 }
2103 \DeclareBoolOption[false]{urluse}
```

`\zxrsetup`

```
2104 \newcommand*{\zxrsetup}{%
2105   \kvsetkeys{ZREF@XR}%
2106 }%
```

`\ZREF@xr@URL`

```
2107 \newcount\ZREF@xr@URL
2108 \ZREF@xr@URL=\ltx@zero
```

`\ZREF@xr@AddURL`

```
2109 \def\ZREF@xr@AddURL#1{%
2110   \begingroup
2111     \def\ZREF@temp{#1}%
2112     \count@=\ltx@one
2113     \ZREF@xr@@AddUrl
2114   \endgroup
2115 }
```

`\ZREF@xr@@AddUrl`

```
2116 \def\ZREF@xr@@AddUrl{%
2117   \ifnum\count@>\ZREF@xr@URL
2118     \global\advance\ZREF@xr@URL by\ltx@one
2119     \xdef\ZREF@xr@theURL{\romannumeral\ZREF@xr@URL}%
2120     \global\expandafter\let
2121         \csname Z@U@\ZREF@xr@theURL\endcsname\ZREF@temp
2122     \@PackageInfo{zref-xr}{%
```

```
      \ltx@backslashchar Z@U@\ZREF@xr@theURL:\MessageBreak
      \ZREF@temp\MessageBreak
    }%
  \else
    \expandafter
    \ifx\csname Z@U@\romannumeral\count@\endcsname\ZREF@temp
      \xdef\ZREF@xr@theURL{\romannumeral\count@}%
    \else
      \expandafter\expandafter\expandafter\ZREF@xr@@AddUrl
    \fi
  \fi
}
```

`\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).

```
\ZREF@IfDefinable\zexternaldocument\def{%
  {%
    \ZREF@NewPropAnchor
    \ZREF@NewPropTitle
    \begingroup
      \csname @safe@actives@true\endcsname
      \makeatletter
      \@ifstar{%
        \ZREF@xr@zreflabelfalse
        \@testopt\ZREF@xr@externaldocument{}%
      }{%
        \ZREF@xr@zreflabeltrue
        \@testopt\ZREF@xr@externaldocument{}%
      }%
  }%
}%
```

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

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

```
\def\ZREF@xr@externaldocument[#1]#2{%
    \def\ZREF@xr@prefix{#1}%
    \let\ZREF@xr@filelist\ltx@empty
    \edef\ZREF@xr@externalfile{#2}%
    \edef\ZREF@xr@file{\ZREF@xr@externalfile.aux}%
    \filename@parse{#2}%
    \@testopt\ZREF@xr@graburl{#2.\zref@xr@ext}%
}%
\def\ZREF@xr@graburl[#1]{%
    \edef\ZREF@xr@url{#1}%
    \ifZREF@xr@urluse
      \expandafter\ZREF@xr@AddURL\expandafter{\ZREF@xr@url}%
      \expandafter\def\expandafter\ZREF@xr@url
      \expandafter{\csname Z@U@\ZREF@xr@theURL\endcsname}%
    \fi
    \ZREF@xr@checkfile
  \endgroup
}%
```

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

```
\def\ZREF@xr@checkfile{%
  \openin\@inputcheck\ZREF@xr@file\relax
```

```
  \ifeof\@inputcheck
    \PackageWarning{zref-xr}{%
      File `\ZREF@xr@file' not found or empty,\MessageBreak
      labels not imported%
    }%
  \else
    \PackageInfo{zref-xr}{%
      Label \ifZREF@xr@zreflabel (zref) \fi
      import from `\ZREF@xr@file'%
    }%
    \def\ZREF@xr@found{0}%
    \def\ZREF@xr@ignored@empty{0}%
    \def\ZREF@xr@ignored@zref{0}%
    \def\ZREF@xr@ignored@ltx{0}%
    \ZREF@xr@processfile
    \closein\@inputcheck
    \begingroup
      \let\on@line\ltx@empty
      \PackageInfo{zref-xr}{%
        Statistics for `\ZREF@xr@file':\MessageBreak
        \ZREF@xr@found\space
        \ifZREF@xr@zreflabel zref\else LaTeX\fi\space
        label(s) found%
        \ifnum\ZREF@xr@ignored@empty>0 %
          ,\MessageBreak
          \ZREF@xr@ignored@empty\space empty label(s) ignored%
        \fi
        \ifnum\ZREF@xr@ignored@zref>0 %
          ,\MessageBreak
          \ZREF@xr@ignored@zref\space
          duplicated zref label(s) ignored%
        \fi
        \ifnum\ZREF@xr@ignored@ltx>0 %
          ,\MessageBreak
          \ZREF@xr@ignored@ltx\space
          duplicated latex label(s) ignored%
        \fi
      }%
    \endgroup
  \fi
  \ifx\ZREF@xr@filelist\ltx@empty
  \else
    \edef\ZREF@xr@file{%
      \expandafter\ltx@car\ZREF@xr@filelist\@nil
    }%
    \edef\ZREF@xr@filelist{%
      \expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil
    }%
    \expandafter\ZREF@xr@checkfile
  \fi
}%
```

`\ZREF@xr@processfile`

```
\def\ZREF@xr@processfile{%
  \read\@inputcheck to\ZREF@xr@line
  \expandafter\ZREF@xr@processline\ZREF@xr@line..\ZREF@nil
  \ifeof\@inputcheck
  \else
    \expandafter\ZREF@xr@processfile
  \fi
}%
```

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

```
\long\def\ZREF@xr@processline#1#2#3\ZREF@nil{%
  \def\x{#1}%
  \toks@{#2}%
  \ifZREF@xr@zreflabel
    \ifx\x\ZREF@xr@zref@newlabel
      \expandafter
      \ZREF@xr@process@zreflabel\ZREF@xr@line...\ZREF@nil
    \fi
  \else
    \ifx\x\ZREF@xr@newlabel
      \expandafter
      \ZREF@xr@process@label\ZREF@xr@line...[]\ZREF@nil
    \fi
  \fi
  \ifx\x\ZREF@xr@@input
    \edef\ZREF@xr@filelist{%
      \etex@unexpanded\expandafter{\ZREF@xr@filelist}%
      {\filename@area\the\toks@}%
    }%
  \fi
}%
\def\ZREF@xr@process@zreflabel\zref@newlabel#1#2#3\ZREF@nil{%
  \edef\ZREF@xr@refname{Z@R@\ZREF@xr@prefix#1}%
  \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
  \def\x{#2}%
  \edef\ZREF@xr@tempname{$temp$}%
  \edef\ZREF@xr@temprefname{Z@R@\ZREF@xr@tempname}%
  \let\ZREF@xr@list\x
  \ifx\ZREF@xr@list\ltx@empty
    \PackageWarningNoLine{zref-xr}{%
      Label `#1' without properties ignored\MessageBreak
      in file `\ZREF@xr@file'%
    }%
    \edef\ZREF@xr@ignored@empty{%
      \the\numexpr\ZREF@xr@ignored@empty+1\relax
    }%
  \else
    \expandafter\ZREF@xr@checklist\x\ZREF@nil
    \expandafter\let\csname\ZREF@xr@temprefname\endcsname\x
    \expandafter\ltx@LocalAppendToMacro
    \csname\ZREF@xr@temprefname\expandafter\endcsname
    \expandafter{%
      \expandafter\externaldocument\expandafter{%
        \ZREF@xr@externalfile
      }%
    }%
    \ZREF@xr@urlcheck\ZREF@xr@tempname
    \ifZREF@xr@tozreflabel
      \@ifundefined{\ZREF@xr@refname}{%
        \ifZREF@xr@verbose
          \PackageInfo{zref-xr}{%
            Import to zref label `\ZREF@xr@tempname#1'%
          }%
        \fi
        \global\expandafter
        \let\csname\ZREF@xr@refname\expandafter\endcsname
        \csname\ZREF@xr@temprefname\endcsname
      }{%
        \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
      }%
    \fi
    \ifZREF@xr@toltxlabel
```

```
      \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
    \fi
  \fi
}%
\def\ZREF@xr@process@label\newlabel#1#2#3[#4]#5\ZREF@nil{%
  \def\ZREF@xr@refname{Z@R@\ZREF@xr@prefix#1}%
  \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
  \def\x{#2}%
  \edef\ZREF@xr@tempname{$temp$}%
  \edef\ZREF@xr@temprefname{Z@R@\ZREF@xr@tempname}%
  \expandafter\ZREF@xr@scanparams
      \csname\ZREF@xr@temprefname\expandafter\endcsname
      \x{}{}{}{}{}\ZREF@nil
  \ifx\\#4\\%
  \else
    % ntheorem knows an optional argument at the end of \newlabel
    \ZREF@NewPropTheotype
    \expandafter\ltx@LocalAppendToMacro
        \csname\ZREF@xr@temprefname\endcsname{\theotype{#4}}%
  \fi
  \expandafter\ltx@LocalAppendToMacro
  \csname\ZREF@xr@temprefname\expandafter\endcsname\expandafter{%
    \expandafter\externaldocument\expandafter{%
      \ZREF@xr@externalfile
    }%
  }%
  \ZREF@xr@urlcheck\ZREF@xr@tempname
  \ifZREF@xr@tozreflabel
    \@ifundefined{\ZREF@xr@refname}{%
      \ifZREF@xr@verbose
        \PackageInfo{zref-xr}{%
          Import to zref label `\ZREF@xr@prefix#1'%
        }%
      \fi
      \global\expandafter
      \let\csname\ZREF@xr@refname\expandafter\endcsname
      \csname\ZREF@xr@temprefname\endcsname
    }{%
      \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
    }%
  \fi
  \ifZREF@xr@toltxlabel
    \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
  \fi
}
\def\ZREF@xr@zref@newlabel{\zref@newlabel}%
\def\ZREF@xr@newlabel{\newlabel}%
\def\ZREF@xr@@input{\@input}%
\def\ZREF@xr@relax{\relax}%
```

\ZREF@xr@tolabel

```
\def\ZREF@xr@tolabel#1#2{%
  \ifZREF@xr@verbose
    \PackageInfo{zref-xr}{%
      Import to LaTeX label `#2'%
    }%
  \fi
  \zref@wrapper@unexpanded{%
    \expandafter\xdef\csname r@#2\endcsname{%
      {%
        \ltx@ifundefined{M@TitleReference}{%
          \ltx@ifundefined{TR@TitleReference}{%
            \zref@extractdefault{#1}{default}{}%
```

```
          }{%
            \noexpand\TR@TitleReference
            {\zref@extractdefault{#1}{default}{}}%
            {\zref@extractdefault{#1}{title}{}}%
          }%
        }{%
          \noexpand\M@TitleReference
          {\zref@extractdefault{#1}{default}{}}%
          {\zref@extractdefault{#1}{title}{}}%
        }%
      }%
      {\zref@extractdefault{#1}{page}{}}%
      \ltx@ifpackageloaded{nameref}{%
        {\zref@extractdefault{#1}{title}{}}%
        {\zref@extractdefault{#1}{anchor}{}}%
        \zref@ifrefcontainsprop{#1}{urluse}{%
          {\zref@extractdefault{#1}{urluse}{}}%
        }{%
          {\zref@extractdefault{#1}{url}{}}%
        }%
      }{}%
    }%
  }%
}
```

\ZREF@xr@zref@ignorewarning

```
\def\ZREF@xr@zref@ignorewarning#1{%
  \PackageWarningNoLine{zref-xr}{%
    Zref label `#1' is already in use\MessageBreak
    in file `\ZREF@xr@file'%
  }%
  \edef\ZREF@xr@ignored@zref{%
    \the\numexpr\ZREF@xr@ignored@zref+1%
  }%
}%
```

\ZREF@xr@ltx@ignorewarning

```
\def\ZREF@xr@ltx@ignorewarning#1{%
  \PackageWarningNoLine{zref-xr}{%
    LaTeX label `#1' is already in use\MessageBreak
    in file `\ZREF@xr@file'%
  }%
  \edef\ZREF@xr@ignored@ltx{%
    \the\numexpr\ZREF@xr@ignored@ltx+1%
  }%
}%
```

\ZREF@xr@checklist

```
\def\ZREF@xr@checklist#1#2#3\ZREF@nil{%
  \ifx\@undefined#1\relax
    \expandafter\ZREF@xr@checkkey\string#1\@nil
  \fi
  \ifx\\#3\\%
  \else
    \ltx@ReturnAfterFi{%
      \ZREF@xr@checklist#3\ZREF@nil
    }%
  \fi
}%
\def\ZREF@xr@checkkey#1#2\@nil{%
  \zref@ifpropundefined{#2}{%
    \zref@newprop{#2}{}%
```

```
  }{}%
}%
```

\ZREF@xr@scanparams

```
\def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{%
  \let#1\ltx@empty
  \ZREF@foundfalse
  \ZREF@xr@scantitleref#1#2\TR@TitleReference{}{}\ZREF@nil
  \ifZREF@found
  \else
    \ltx@LocalAppendToMacro#1{\default{#2}}%
  \fi
  % page
  \ltx@LocalAppendToMacro#1{\page{#3}}%
  % nameref title
  \ifZREF@found
  \else
    \ifx\\#4\\%
    \else
      \def\ZREF@xr@temp{#4}%
      \ifx\ZREF@xr@temp\ZREF@xr@relax
      \else
        \ltx@LocalAppendToMacro#1{\title{#4}}%
      \fi
    \fi
  \fi
  % anchor
  \ifx\\#5\\%
  \else
    \ltx@LocalAppendToMacro#1{\anchor{#5}}%
  \fi
  \ifx\\#6\\%
  \else
    \ifZREF@xr@urluse
      \ZREF@xr@AddURL{#6}%
      \expandafter\ltx@LocalAppendToMacro\expandafter#1%
      \expandafter{%
        \expandafter\urluse\expandafter{%
          \csname Z@U@\ZREF@xr@theURL\endcsname
        }%
      }%
    \else
      \ltx@LocalAppendToMacro#1{\url{#6}}%
    \fi
  \fi
}%
```

\ZREF@xr@scantitleref

```
\def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{%
  \ifx\\#5\\%
  \else
    \ltx@LocalAppendToMacro#1{%
      \default{#3}%
      \title{#4}%
    }%
    \ZREF@foundtrue
  \fi
}%
```

\ZREF@xr@urlcheck

```
\def\ZREF@xr@urlcheck#1{%
  \zref@ifrefcontainsprop{#1}{anchor}{%
```

```
    \zref@ifrefcontainsprop{#1}{url}{%
    }{%
      \expandafter
      \ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
      \expandafter{%
        \csname url\ifZREF@xr@urluse use\fi
        \expandafter\endcsname\expandafter{\ZREF@xr@url}%
      }%
    }%
  }{%
  }%
}%
```

⟨/xr⟩

6.17 Module **hyperref**

UNFINISHED :-(

```
⟨*hyperref⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-hyperref}%
  [2025-06-08 v2.36 Module hyperref for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\ZREF@NewPropAnchor
\zref@addprop\ZREF@mainlist{anchor}%
⟨/hyperref⟩
```

6.18 Module **savepos**

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

6.18.1 Identification

```
⟨*savepos⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-savepos}%
  [2025-06-08 v2.36 Module savepos for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
```

6.18.2 Availability

We now assume that it is available always.

6.18.3 Setup

```
\zref@newlist{savepos}
\ExplSyntaxOn
  \zref@newprop*{posx}[0]{\int_use:N \tex_lastxpos:D}
  \zref@newprop*{posy}[0]{\int_use:N \tex_lastypos:D}
\zref@addprops{savepos}{posx,posy}
```

6.18.4 User macros

`\zref@savepos`

```
  \def\zref@savepos{%
    \if@filesw
      \tex_savepos:D
    \fi}
\ExplSyntaxOff
```

\ZREF@zsavepos

```
\def\ZREF@zsavepos#1#2#3{%
  \@bsphack
  \if@filesw
    \zref@savepos
    #1{#3}{#2}%
    \zref@savepos
  \fi
  \@esphack
}
```

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

```
\ZREF@IfDefinable\zsavepos\def{%
  {%
    \ZREF@zsavepos\zref@labelbylist{savepos}%
  }%
}
```

\zsaveposx

```
\ZREF@IfDefinable\zsaveposx\def{%
  {%
    \ZREF@zsavepos\zref@labelbyprops{posx}%
  }%
}
```

\zsaveposy

```
\ZREF@IfDefinable\zsaveposy\def{%
  {%
    \ZREF@zsavepos\zref@labelbyprops{posy}%
  }%
}
```

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

```
\newcommand*{\zposx}[1]{%
  \zref@extract{#1}{posx}%
}%
\newcommand*{\zposy}[1]{%
  \zref@extract{#1}{posy}%
}%
```

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

Also it is in the responsibility of the user to marked used positions by \zrefused in order to notify LATEX about undefined references.

\ZREF@savepos@ok

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

⟨/savepos⟩

6.19 Module **abspos**

6.19.1 Identification

```
⟨*abspos⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-abspos}%
  [2025-06-08 v2.36 Module abspos for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\RequirePackage{zref-savepos}[2019/11/29]
\ifx\ZREF@savepos@ok Y%
\else
  \expandafter\endinput
\fi
\RequirePackage{zref-pagelayout}[2019/11/29]
\zref@addprop{savepos}{abspage}
\zref@addprop{savepos}{pagevalue}
```

\zref@absposx

```
\newcommand*{\zref@absposx}[3]{%
  \number
    \expandafter\zref@absposnumx\expandafter{%
      \number\zref@extractdefault{#1}{abspage}{0}%
    }{#2}{#3}%
  \ltx@space
}
```

\zref@absposy

```
\newcommand*{\zref@absposy}[3]{%
  \number
  \expandafter\zref@absposnumy\expandafter{%
    \number\zref@extractdefault{#1}{abspage}{0}%
  }{#2}{#3}%
  \ltx@space
}
```

\zref@absposnumx

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

\zref@absposnumy

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

\ZREF@absposnum

```
\def\ZREF@absposnum#1#2#3#4{%
  \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
    0%
  }{%
    \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
  }%
}
```

```
\zref@def@absposx
2587 \ZREF@Robust\def\zref@def@absposx#1{%
2588   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposx}%
2589 }

\zref@def@absposy
2590 \ZREF@Robust\def\zref@def@absposy#1{%
2591   \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
2592 }

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

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

\ZREF@def@abspos
2599 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2600   \edef#1{#2{#3}{#4}{#5}}%
2601 }

\zref@absposused
2602 \ZREF@Robust\def\zref@absposused{%
2603   \zref@wrapper@babel\ZREF@abspos@used
2604 }

\ZREF@abspos@used
2605 \def\ZREF@abspos@used#1{%
2606   \zref@refused{#1}%
2607   \zref@ifrefundefined{#1}{%
2608   }{%
2609     \begingroup
2610       \edef\ZREF@temp{%
2611         \zref@extractdefault{#1}{abspage}{0}%
2612       }%
2613       \ifnum\ZREF@temp>\ltx@zero
2614         \zref@refused{thepage\ZREF@temp}%
2615       \else
2616         \@PackageError{zref-abspos}{%
2617           \string\zref@pos@label@used\ltx@space
2618           needs property `abspage'\MessageBreak
2619           in label `#1'%
2620         }\@ehc
2621       \fi
2622     \endgroup
2623   }%
2624 }

\zref@absposnumused
2625 \newcommand*{\zref@absposnumused}[1]{%
2626   \ifnum#1>\ltx@zero
2627     \zref@refused{thepage\number#1}%
2628   \else
2629     \@PackageError{zref-abspos}{%
2630       Invalid absolute page number (#1)\MessageBreak
2631       for \string\zref@pos@num@used.\MessageBreak
2632       A positive integer is expected%
```

```
    }\@ehc
  \fi
}
```

\zref@ifabsposundefined

```
\def\zref@ifabsposundefined#1{%
  \zref@ifrefundefined{#1}\ltx@firsttwo{%
    \expandafter\zref@ifabsposnumundefined\expandafter{%
      \number\zref@extractdefault{#1}{abspage}{0}%
    }%
  }%
}
```

\zref@ifabsposnumundefined

```
\def\zref@ifabsposnumundefined#1{%
  \ifnum\ZREF@number{#1}>\ltx@zero
    \zref@ifrefundefined{thepage#1}%
        \ltx@firstoftwo\ltx@secondoftwo
  \else
    \expandafter\ltx@firstoftwo
  \fi
}
```

6.19.2 Media

Ensure that \stockwidth and \stockheight are defined

```
\@ifundefined{stockwidth}{\newdimen\stockwidth}{}
\@ifundefined{stockheight}{\newdimen\stockheight}{}
```

\ZREF@abspos@media@width

```
\edef\ZREF@abspos@media@width{%
  \ltx@ifundefined{pdfpagewidth}{%
    \ltx@ifundefined{mediawidth}{%
      \ifdim\stockwidth > \z@
        stockwidth%
      \else
        paperwidth%
      \fi
    }{%
      mediawidth%
    }%
  }{%
    pdfpagewidth%
  }%
}
\ifluatex
\def\ZREF@abspos@media@width{pdfpagewidth}%
\fi
```

\ZREF@abspos@media@height

```
\edef\ZREF@abspos@media@height{%
  \ltx@ifundefined{pdfpageheight}{%
    \ltx@ifundefined{mediaheight}{%
      \ifdim\stockwidth > \z@
        stockheight%
      \else
        paperheight%
      \fi
    }{%
      mediaheight%
    }%
  }{%
```

```
    \noexpand\ifcase\pdfpageheight
      \ifdim\stockwidth > \z@
        stockheight%
      \else
        paperheight%
      \fi
    \noexpand\else
      pdfpageheight%
    \noexpand\fi
  }%
}
\ifluatex
\edef\ZREF@abspos@media@height{%
   \noexpand\ifcase\pageheight
     \ifdim\stockwidth > \z@
       stockheight%
     \else
       paperheight%
     \fi
    \noexpand\else
      pdfpageheight%
    \noexpand\fi}%
\fi
```

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

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

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

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

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

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

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

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

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

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

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

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

6.19.3 Paper

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

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

```
\ZREF@abspos@paper@x@right
                        \def\ZREF@abspos@paper@x@right#1{%
                          \zref@extract{#1}{paperwidth}%
                        }
\ZREF@abspos@paper@x@center
                        \def\ZREF@abspos@paper@x@center#1{%
                          \zref@extract{#1}{paperwidth}/2%
                        }
  \ZREF@abspos@paper@y@top
                        \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top
\ZREF@abspos@paper@y@bottom
                        \def\ZREF@abspos@paper@y@bottom#1{%
                          \ZREF@abspos@paper@y@top{#1}%
                          -\zref@extract{#1}{paperheight}%
                        }
\ZREF@abspos@paper@y@center
                        \def\ZREF@abspos@paper@y@center#1{%
                          \ZREF@abspos@paper@y@top{#1}%
                          -\zref@extract{#1}{paperheight}/2%
                        }
```

6.19.4 Origin

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

```
    \ZREF@abspos@origin@x
                        \let\ZREF@temp\ltx@two
                        \ltx@IfUndefined{pdfhorigin}{}{%
                        %  \ifpdf
                            \let\ZREF@temp\ltx@zero
                        %  \fi
                        }
                        \ifluatex
                        % \ifpdf
                          \let\ZREF@temp\ltx@zero
                        % \fi
                        \fi

                        \ifx\ZREF@temp\ltx@two
                          \ifnum\mag=1000 %
                            \let\ZREF@temp\ltx@one
                          \fi
                        \fi
                        \ifcase\ZREF@temp
                          \def\ZREF@abspos@origin@x#1{%
                            \zref@extract{#1}{pdfhorigin}%
                          }%
                        \or
                          \def\ZREF@abspos@origin@x#1{%
                            4736286%
                          }%
                        \or
                          \def\ZREF@abspos@origin@x#1{%
                            \numexpr\mag/1000*\dimexpr 1truein\relax\relax
                          }%
                        \fi
```

\ZREF@abspos@origin@y

```
\let\ZREF@temp\ltx@two
\ltx@IfUndefined{pdfvorigin}{}{%
%  \ifpdf
    \let\ZREF@temp\ltx@zero
%  \fi
}
\ifluatex
%  \ifpdf
    \let\ZREF@temp\ltx@zero
%  \fi
\fi
\ifx\ZREF@temp\ltx@two
  \ifnum\mag=1000 %
    \let\ZREF@temp\ltx@one
  \fi
\fi
\ifcase\ZREF@temp
  \def\ZREF@abspos@origin@y#1{%
    \zref@extract{#1}{pdfvorigin}%
  }%
\or
  \def\ZREF@abspos@origin@y#1{%
    4736286%
  }%
\or
  \def\ZREF@abspos@origin@y#1{%
    \numexpr\mag/1000*\dimexpr 1truein\relax\relax
   }%
\fi
```

6.19.5 Header

\ZREF@abspos@head@x@left

```
\def\ZREF@abspos@head@x@left#1{%
  \ZREF@abspos@paper@x@left{#1}%
  +\ZREF@abspos@origin@x{#1}%
  +\zref@extract{#1}{hoffset}%
  +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
     \zref@extract{#1}{oddsidemargin}%
   \else
     \zref@extract{#1}{evensidemargin}%
   \fi
}
```

\ZREF@abspos@head@x@right

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

\ZREF@abspos@head@x@center

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

\ZREF@abspos@head@y@top

```
\def\ZREF@abspos@head@y@top#1{%
  \ZREF@abspos@paper@y@top{#1}%
  -\ZREF@abspos@origin@y{#1}%
  -\zref@extract{#1}{voffset}%
```

```
2824   -\zref@extract{#1}{topmargin}%
2825 }
```

\ZREF@abspos@head@y@bottom

```
2826 \def\ZREF@abspos@head@y@bottom#1{%
2827   \ZREF@abspos@head@y@top{#1}%
2828   -\zref@extract{#1}{headheight}%
2829 }
```

\ZREF@abspos@head@y@center

```
2830 \def\ZREF@abspos@head@y@center#1{%
2831   \ZREF@abspos@head@y@top{#1}%
2832   -\zref@extract{#1}{headheight}/2%
2833 }
```

6.19.6 Body

\ZREF@abspos@body@x@left

```
2834 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left
```

\ZREF@abspos@body@x@right

```
2835 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right
```

\ZREF@abspos@body@x@center

```
2836 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center
```

\ZREF@abspos@body@y@top

```
2837 \def\ZREF@abspos@body@y@top#1{%
2838   \ZREF@abspos@head@y@bottom{#1}%
2839   -\zref@extract{#1}{headsep}%
2840 }
```

\ZREF@abspos@body@y@bottom

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

\ZREF@abspos@body@y@center

```
2845 \def\ZREF@abspos@body@y@center#1{%
2846   \ZREF@abspos@body@y@top{#1}%
2847   -\zref@extract{#1}{textheight}/2%
2848 }
```

6.19.7 Footer

\ZREF@abspos@foot@x@left

```
2849 \let\ZREF@abspos@foot@x@left\ZREF@abspos@head@x@left
```

\ZREF@abspos@foot@x@right

```
2850 \let\ZREF@abspos@foot@x@right\ZREF@abspos@head@x@right
```

\ZREF@abspos@foot@x@center

```
2851 \let\ZREF@abspos@foot@x@center\ZREF@abspos@head@x@center
```

\ZREF@abspos@foot@y@bottom

```
2852 \def\ZREF@abspos@foot@y@bottom#1{%
2853   \ZREF@abspos@body@y@bottom{#1}%
2854   -\zref@extract{#1}{footskip}%
2855 }
```

6.19.8 Marginal notes

\ZREF@abspos@marginpar@x@left

```
\def\ZREF@abspos@marginpar@x@left#1{%
  \ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
    \ZREF@abspos@body@x@right{#1}%
    +\zref@extract{#1}{marginparsep}%
  \else
    \ZREF@abspos@body@x@left{#1}%
    -\zref@extract{#1}{marginparsep}%
    -\zref@extract{#1}{marginparwidth}%
  \fi
}
```

\ZREF@abspos@marginpar@x@right

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

ZREF@abspos@marginpar@x@center

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

\ZREF@abspos@marginpar@y@top

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

ZREF@abspos@marginpar@y@bottom

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

ZREF@abspos@marginpar@y@center

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

6.19.9 Stock paper

\ZREF@abspos@stock@x@left

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

\ZREF@abspos@stock@x@right

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

\ZREF@abspos@stock@x@center

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

\ZREF@abspos@stock@y@top

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

\ZREF@abspos@stock@y@bottom

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

\ZREF@abspos@stock@y@center

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

⟨/abspos⟩

6.20 Module dotfill

```
⟨*dotfill⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-dotfill}%
  [2025-06-08 v2.36 Module dotfill for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
```

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

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

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

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

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

```
\RequirePackage{keyval}
```

The definitions of the keys follow.

```
\define@key{ZREF@DF}{unit}{%
  \def\ZREF@df@unit{#1}%
}
\define@key{ZREF@DF}{min}{%
  \def\ZREF@df@min{#1}%
}
\define@key{ZREF@DF}{dot}{%
  \def\ZREF@df@dot{#1}%
}
```

Defaults are set, see user interface.

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

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

```
\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.

```
\ZREF@IfDefinable\zdotfill\def{%
  {%
    \leavevmode
    \global\advance\c@zref@unique\ltx@one
    \begingroup
      \def\ZREF@temp{zref@\number\c@zref@unique}%
      \csname tex_savepos:D\endcsname
      \zref@labelbyprops{\thezref@unique L}{posx}%
      \setlength{\dimen@}{\ZREF@df@unit}%
      \zref@ifrefundefined{\thezref@unique R}{%
        \ZREF@dotfill
      }{%
        \ifnum\numexpr\zposx{\thezref@unique R}%
                      -\zposx{\thezref@unique L}\relax
            <\dimexpr\ZREF@df@min\dimen@\relax
          \hfill
        \else
          \ZREF@dotfill
        \fi
      }%
      \csname tex_savepos:D\endcsname
      \zref@labelbyprops{\thezref@unique R}{posx}%
```

```
    \endgroup
    \kern\z@
  }%
}
```

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

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

⟨/dotfill⟩

6.21 Module **env**

```
⟨*env⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-env}%
  [2025-06-08 v2.36 Module env for zref (HO)]%
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
  \expandafter\endinput
\fi
\zref@newprop{envname}[]{\@currenvir}
\zref@newprop{envline}[]{\zref@env@line}
```

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

```
\def\zref@env@line{%
  \ifx\@currenvline\ltx@empty
  \else
    \expandafter
    \ZREF@ENV@line\@currenvline\ltx@empty line \ltx@empty\@nil
  \fi
}
```

`\ZREF@ENV@line`

```
\def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%
```

⟨/env⟩

7 Installation

7.1 Download

Package. This package is available on CTAN[2]:

CTAN:macros/latex/contrib/zref/zref.dtx The source file.

CTAN:macros/latex/contrib/zref/zref.pdf Documentation.

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

CTAN:install/macros/latex/contrib/zref.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

[2] CTAN:pkg/zref

7.2 Bundle installation

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

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

7.3 Package installation

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

```
tex zref.dtx
```

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

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

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

7.4 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

7.5 Some details for the interested

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

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

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

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

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

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

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

An example follows how to generate the documentation with pdfLaTeX:

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

8 References

[1] Package footmisc, Robin Fairbairns, 2004/01/23 v5.3a. CTAN:pkg/footmisc

[2] Package hyperref, Sebastian Rahtz, Heiko Oberdiek, 2006/08/16 v6.75c. CTAN:pkg/hyperref

[3] Package lastpage, Jeff Goldberg, 1994/06/25 v0.1b. CTAN:pkg/lastpage

[4] Package nameref, Sebastian Rahtz, Heiko Oberdiek, 2006/02/12 v2.24. CTAN:pkg/nameref

[5] Package perpage, David Kastrup, 2002/12/20 v1.0. CTAN:pkg/perpage

[6] Package titleref, Donald Arsenau, 2001/04/05 v3.1. CTAN:pkg/titleref

[7] Package totpages, Wilhelm Müller, 1999/07/14 v1.00. CTAN:pkg/totpages

[8] Package xr, David Carlisle, 1994/05/28 v5.02. CTAN:pkg/xr

[9] Package xr-hyper, David Carlisle, 2000/03/22 v6.00beta4. CTAN:pkg/xr-hyper

9 History

[2006/02/20 v1.0]

- First version.

[2006/05/03 v1.1]

- Module perpage added.
- Module redesign as packages.

[2006/05/25 v1.2]

- Module dotfillmin added.
- Module base: macros `\zref@require@unique` and `\thezref@unique` added (used by modules titleref and dotfillmin).

[2006/09/08 v1.3]

- Typo fixes and English cleanup by Per Starback.

[2007/01/23 v1.4]

- Typo in macro name fixed in documentation.

[2007/02/18 v1.5]

- `\zref@getcurrent` added (suggestion of Igor Akkerman).
- Module savepos also supports XeTeX.

[2007/04/06 v1.6]

- Fix in modules abspage and base: Now counter `abspage` and `zref@unique` are not remembered by `\include`.
- Beamer support for module titleref.

[2007/04/17 v1.7]

- Package atbegshi replaces everyshi.

[2007/04/22 v1.8]

- `\zref@wrapper@babel` and `\zref@refused` are now expandable if babel is not used or `\if@safe@actives` is already set to true. (Feature request of Josselin Noirel)

[2007/05/02 v1.9]

- Module titleref: Some support for `\caption` of package longtable, but only if `\label` is given after `\caption`.

[2007/05/06 v2.0]

- Uses package etexcmds for accessing ε-TeX's `\unexpanded`.

[2007/05/28 v2.1]

- Module titleref supports caption of package listings.
- Fixes in module titleref for support of packages titlesec and longtable.

[2008/09/21 v2.2]

- Module base: `\zref@iflistcontainsprop` is documented, but a broken `\zref@listcontainsprop` implemented. Name and implementation fixed (thanks Ohad Kammar).

[2008/10/01 v2.3]

- `\zref@localaddprop` added (feature request of Ohad Kammar).
- Module lastpage: list 'LastPage' added. Label 'LastPage' will use the properties of this list (default is empty) along with the properties of the main list.

[2009/08/07 v2.4]

- Module runs added.

[2009/12/06 v2.5]

- Module lastpage: Uses package atveryend.
- Module titleref: Further commands are disabled during string expansion, imported from package nameref.

[2009/12/07 v2.6]

- Version date added for package atveryend.

[2009/12/08 v2.7]

- Module titleref: Use of package gettitlestring.

[2010/03/26 v2.8]

- `\zifrefundefined` added.
- Module lastpage: Macros `\zref@iflastpage` and `\ziflastpage` added.
- Module thepage added.
- Module nextpage added.

[2010/03/29 v2.9]

- Module marks added (without documentation).
- `\zref@addprop` now adds expanded property to list.
- Useless `\ZREF@ErrorNoLine` removed.

[2010/04/08 v2.10]

- Module xr remembers the external document name in property 'externaldocument'.

[2010/04/15 v2.11]

- Module titleref: Better support of class memoir.
- Module titleref: Support of theorems.

[2010/04/17 v2.12]

- Module base: `\zref@newprop` ensures global empty default.
- Module xr: Setup options tozreflabel and toltxlabel added.

[2010/04/19 v2.13]

- `\zref@setcurrent` throws an error if the property does not exist (Florent Chervet).
- `\zref@getcurrent` the documentation is fixed (Florent Chervet). Also it returns the empty string in case of errors.
- `\zref@addprop` and `\zref@localaddprop` now take a list of property names (feature request of Florent Chervet).
- Example for `\zref@wrapper@unexpanded` corrected (Florent Chervet).

[2010/04/22 v2.14]

- Bug fix for `\zref@getcurrent` second argument wasn't eaten in case of unknown property.
- `\zref@getcurrent` supports `\zref@wrapper@unexpanded`.
- `\zref@wrapper@unexpanded` added for `\ZREF@xr@tolabel`.
- `\zref@extract`, `\zref@extractdefault`, `\zref@getcurrent` are expandable in exact two steps except inside `\zref@wrapper@unexpanded`.

[2010/04/23 v2.15]

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

[2010/04/28 v2.16]

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

[2010/05/01 v2.17]

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

[2010/05/13 v2.18]

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

[2010/10/22 v2.19]

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

[2011/02/12 v2.20]

- Fix for warning in zref-xr.

[2011/03/18 v2.21]

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

[2011/10/05 v2.22]

- Documentation fixed for `\zref@(local)addprop(s)`.
- Module base: `\zref@def@extract`, `\zref@def@extractdefault` added.
- Fix in module pagelayout: Because of missing `\noexpand` commands the values of the pagelayout properties on all pages were the values at package loading.
- Module base: `\zref@showprop` added.

[2011/12/05 v2.23]

- Module savepos: `\zsaveposx` and `\zsaveposy` added.

[2012/04/04 v2.24]

- Module titleref, package titlesec: some support for class 'straight' (`\ttl@straight@i`) added.

[2016/05/16 v2.25]

- Documentation updates.

[2016/05/21 v2.26]

- update zref-savepos for new luatex

[2018/11/21 v2.27]

- adapted zref-perpage, see issue https://github.com/ho-tex/zref/issues/2

[2019/11/29 v2.28]

- Documentation updates.
- Use iftex directly.

[2020-03-03 v2.29]

- adapted in module zref-pagelayout the properties pdfhorigin, pdfvorigin, pdfpagewidth, pdfpageheight for luatex to the right primitives.
- Removed no longer needed code for older lualatex versions.
- added some documentation of the abspos module.
- adapted the abspos module to the new luatex primitives.
- adapted pageattr module to the new luatex primitives.
- added short documentation for pageattr module
- use luatex command names directly in zref-savepos rather than defining pdftex compatibility names.
- allow zref-abspos to use `\pdf[vh]origin` in dvi mode.

[2020-03-04 v2.30]

- add pagevalue property to savepos in the abspos module (issue 1)

[2020-05-28 v2.31]

- Adapted module zref-counter to use `\@currentcounter` in the next LaTeXversion (issue 5)

[2020-07-03 v2.32]

- Changed in zref-pagelayout the names of the shipout box dimensions to adapt to the new hook management.

[2022-03-08 v2.33]

- Avoid that amstext undoes the stepcounter patch in zref-perpage, https://github.com/ho-tex/zref/issues/11
- Make the unique counter more robust when includeonly is used, https://github.com/ho-tex/zref/issues/10

[2022-04-07 v2.34]

- use the `zref@unique` counter in the include hook only if the module needs it https://github.com/ho-tex/zref/issues/14

[2023-09-14 v.2.35]

- Define the abspage counter only if it is undefined https://github.com/ho-tex/zref/issues/14
- Remove dependency from atveryend.

[2025-06-08 v2.36]

- use `\tex_savepos:D` for better engine compability, https://github.com/ho-tex/zref/issues/20
- adapt lastpage code to format change. https://github.com/ho-tex/zref/issues/22

10 Index

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

Symbols

\@PackageError .. 509, 525, 2616, 2629
\@PackageInfo 2122
\@PackageInfoNoLine 549, 564, 1368, 1463, 1475, 1542, 1594
\@PackageWarning 692
\@addtoreset 1014
\@auxout 716
\@begintheorem 2061, 2066
\@bsphack 601, 611, 631, 2510
\@caption 1906
\@chapter 1918, 1955
\@currentHref 945
\@currentcounter 1036
\@currentlabel 940
\@currenvir 2948
\@currenvline 2951, 2954
\@ehc 297, 307, 492, 515, 527, 1626, 2620, 2633
\@elt 1749
\@esphack 608, 628, 643, 2516
\@firstofone 1582
\@firstoftwo 1448, 1526
\@ifclassloaded 1953, 1990
\@ifdefinable 243, 290
\@ifl@t@r 223, 1034, 1063
\@ifnextchar 531, 1772
\@ifpackageloaded 2010, 2029, 2037, 2056
\@ifstar 496, 2142
\@ifundefined 192, 910, 1012, 1783, 2278, 2320, 2651, 2652
\@input 2339
\@inputcheck 2170, 2171, 2186, 2223, 2225
\@latex@warning 778
\@mainaux 1711
\@namedef 536
\@ne 1748
\@newl@bel 286
\@nil 1641, 1643, 1871, 1878, 2060, 2061, 2214, 2217, 2397, 2406, 2954, 2957
\@onelevel@sanitize 423, 441, 506, 534, 2059, 2062
\@opargbegintheorem 2051
\@part 1912
\@schapter 1936
\@secondoftwo 1450, 1528
\@sect 1924
\@spart 1930
\@ssect 1942
\@stpelt 1749, 1759, 1760
\@testdef 1350, 1351, 1510, 1511
\@testopt 2144, 2147, 2157
\@tfor 323, 721
\@undefined 1827, 2396
\\ 25, 26, 27, 28, 153, 155, 157, 158, 170, 173, 2305, 2399, 2424, 2434, 2438, 2454
\␣ 44, 45

A

\AddLineBeginAux 281
\addtocounter 1748
\AddToHook . 917, 1019, 1064, 1075, 1122, 1359, 1432, 1519, 1603, 1742
\advance 1078, 1415, 1571, 1796, 2118, 2912
\afterassignment 234, 1162, 1166
\Alph 7
\anchor 2436
\AtBeginDocument 1042, 1251, 1705, 1904
\AtBeginShipoutAddToBox 1123
\AtBeginShipoutBoxDepth 1339
\AtBeginShipoutBoxHeight 1338
\AtBeginShipoutBoxWidth 1337
\AtEndOfPackage 195

B

\beamer@section 1992
\beamer@subsection 1998
\beamer@subsubsection 2004
\begin . 23, 57, 100, 106, 156, 172, 1758
\bfseries 933

C

\c@abspage 1804, 1811
\c@page ... 959, 1078, 1801, 2806, 2857
\c@zpage 1801, 1807
\c@zref@unique 913, 920, 1796, 2912, 2914
\ch@pt@c 1959, 1965, 1971
\chapter 24, 30, 32, 61, 82
\ChapterPages 91, 112
\ChapterStart 78, 135, 150, 166
\ChapterStop 85, 148, 165, 184
\chardef 1184, 1199, 1208, 1212
\cleaders 2936
\cleardoublepage 79, 86
\clearpage 62
\closein 2186
\columnsep 1323
\columnwidth 1322
\comma@entry ... 344, 345, 347, 353, 379, 380, 382, 388, 615, 617, 621, 1647, 1648, 1649, 1655, 1658
\comma@parse 343, 378, 614, 1646
\count@ 1365, 1376, 1377, 1379, 1414, 1415, 1424, 1426, 1427, 1539, 1562, 1564, 1565, 1570, 1571, 2112, 2117, 2128, 2129
\cs 1756

\csname 253, 254, 291, 316, 317, 318, 327, 352, 353, 370, 371, 387, 388, 405, 406, 426, 428, 445, 463, 466, 479, 537, 539, 540, 545, 555, 560, 566, 579, 588, 605, 621, 652, 662, 731, 739, 745, 788, 789, 791, 811, 834, 835, 836, 897, 920, 1355, 1515, 1676, 1684, 1736, 1746, 1749, 1784, 1786, 1789, 1791, 1803, 1809, 1814, 1815, 1817, 1819, 1820, 1827, 1880, 1883, 2121, 2128, 2140, 2164, 2268, 2270, 2285, 2286, 2303, 2310, 2313, 2327, 2328, 2348, 2445, 2468, 2470, 2584, 2915, 2929
\current@chapid 80, 88

D

\DeclareBoolOption 2097, 2098, 2099, 2103
\DeclareOption 194
\default 2417, 2457
\define@key 1879, 1882, 1885, 1888, 2100, 2896, 2899, 2902
\descriptionlabel 1948
\detokenize 1866
\dftest 167, 174, 175, 176, 177, 178, 179, 180, 181, 182
\dimen@ 2917, 2923, 2936
\dimexpr 153, 155, 1427, 2770, 2799, 2923
\do 328, 721
\documentclass 4, 39, 68, 273
\dotfill 169, 173

E

\emph 150
\end . . . 34, 64, 130, 159, 183, 185, 1753
\endcsname 253, 254, 291, 316, 317, 318, 327, 352, 353, 370, 371, 387, 388, 405, 406, 426, 428, 445, 464, 466, 479, 537, 539, 540, 545, 555, 560, 566, 579, 588, 605, 621, 652, 662, 737, 739, 745, 788, 789, 791, 811, 834, 835, 836, 884, 897, 920, 1355, 1515, 1676, 1684, 1735, 1736, 1745, 1746, 1749, 1761, 1784, 1786, 1789, 1790, 1791, 1803, 1809, 1814, 1815, 1817, 1819, 1820, 1827, 1880, 1883, 2121, 2128, 2140, 2164, 2268, 2270, 2285, 2286, 2303, 2310, 2313, 2327, 2328, 2348, 2445, 2468, 2471, 2584, 2915, 2929
\endinput 192, 265, 278, 971, 1009, 1032, 1060, 1113, 1153, 1244, 1262, 1443, 1488, 1614, 1725, 1838, 2082, 2485, 2497, 2548, 2553, 2891, 2946
\escapechar 330, 421, 461, 462, 468, 723, 1268
\etex@unexpanded . 592, 820, 840, 2246
\evensidemargin 1314
\ExplSyntaxOff 2508
\ExplSyntaxOn 2500
\externaldocument 2272, 2314

F

\fancyhead 51, 54
\fancyhf 50, 53
\fancypagestyle 52
\filename@area 2247
\filename@parse 2156
\fmtversion 223, 1034, 1063
\foo 18, 29, 31, 33
\footskip 1319
\foremargin 1326
\frontmatter 58, 103

G

\g@addto@macro 351, 369, 1789
\G@refundefinedtrue 777
\gdef 413, 540, 545, 950, 1355, 1712, 1784, 1786
\GetTitleStringDisableCommands 1875
\GetTitleStringExpand 1861
\GetTitleStringNonExpand 1863
\GetTitleStringResult 1866

H

\hb@xt@ 2936
\headheight 1317
\headmargin 1329
\headsep 1318
\hfill 2924, 2936
\hoffset 1310
\hss 2936

I

\if@filesw 711, 1065, 1076, 1710, 2505, 2511
\if@safe@actives 890
\ifcase 115, 1220, 1647, 2683, 2696, 2760, 2789
\ifcsname . 884, 1735, 1745, 1761, 1790
\ifdim 2656, 2674, 2684, 2697
\ifeof 2171, 2225
\ifetex@unexpanded 268
\iffirstchoice@ 1744
\IfFormatAtLeastTF 223, 915, 1330, 1740
\ifheadnameref 1967, 1980
\ifin@ 318
\ifluatex . 1287, 1447, 1524, 1537, 1580, 1596, 2668, 2694, 2749, 2779
\ifnum 477, 1090, 1195, 1205, 1211, 1414, 1570, 1619, 1672, 2117, 2194, 2198, 2203, 2613, 2626, 2644, 2756, 2785, 2921
\ifodd 124, 2806, 2857
\ifpdf 2745, 2750, 2775, 2780
\ifx 438, 442, 475, 508, 566, 674, 677, 691, 730, 796, 969, 974, 981, 1007, 1030, 1058, 1111, 1151, 1242, 1260, 1354, 1441, 1487, 1495, 1514,

1525, 1581, 1612, 1629, 1638, 1642, 1647, 1648, 1649, 1723, 1814, 1836, 1959, 1963, 2020, 2040, 2064, 2080, 2128, 2211, 2234, 2239, 2244, 2258, 2305, 2396, 2399, 2424, 2427, 2434, 2438, 2454, 2483, 2495, 2546, 2551, 2755, 2784, 2889, 2944, 2951
\ifZREF@found 248, 2415, 2422
\ifZREF@immediate 635, 701, 713, 717, 732
\ifZREF@pa@list 1502, 1507
\ifZREF@pl@list 1342, 1347
\ifzref@titleref@expand . 1844, 1860
\ifzref@titleref@stripperiod 1857, 1868
\ifZREF@xr@toltxlabel . . . 2291, 2333
\ifZREF@xr@tozreflabel . . 2277, 2319
\ifZREF@xr@urluse . . 2161, 2440, 2470
\ifZREF@xr@verbose . . 2279, 2321, 2342
\ifZREF@xr@zreflabel 2092, 2178, 2192, 2233
\immediate 706, 1711
\in@ 315
\int 2501, 2502
\item 107, 111, 113, 121, 125, 127

K

\kern 2932
\kv@define@key 645, 656, 667, 672, 689
\kv@key . . . 693, 1641, 1643, 1644, 1658
\kv@parse 690, 1637
\kv@value 691, 1638, 1639, 1646
\kvsetkeys 634, 1892, 2105, 2908

L

\label 974, 981, 1900
\leavevmode 2911
\lst@@caption 2043
\lst@label 2040
\lst@MakeCaption 2039
\LT@c@ption 2031
\ltx@backslashchar 744, 1536, 1595, 2123
\ltx@car 1641, 2214
\ltx@cdr 1643, 2217
\ltx@empty 291, 500, 566, 633, 720, 797, 1120, 1629, 1845, 1871, 1878, 1959, 1963, 2020, 2040, 2153, 2188, 2211, 2217, 2258, 2412, 2951, 2954, 2957
\ltx@firstofone 255, 868, 879, 885, 1496, 1497
\ltx@firstoftwo 800, 827, 828, 893, 1092, 2646, 2648
\ltx@firsttwo 2637
\ltx@gobble 251, 356, 391, 624, 664, 974, 975, 981, 1417, 1573, 1660, 1900
\ltx@gobblethree 982
\ltx@gobbletwo . 695, 1014, 1662, 1766
\ltx@ifpackageloaded 2365
\ltx@IfUndefined 230, 250, 258, 411, 878, 924, 1128, 1462, 1474, 1493, 1494, 1523, 1579, 1954, 1956, 2744, 2774
\ltx@ifundefined 301, 486, 584, 761, 806, 945, 1269, 1349, 1509, 2090, 2350, 2351, 2581, 2654, 2655, 2672, 2673
\ltx@LocalAppendToMacro 386, 404, 650, 660, 1593, 2269, 2309, 2312, 2417, 2420, 2429, 2436, 2442, 2449, 2456, 2468
\ltx@newif 1342, 1502
\ltx@one 1415, 1571, 1796, 2112, 2118, 2757, 2786, 2912
\ltx@onelevel@sanitize 558, 563
\ltx@ReturnAfterFi 2401
\ltx@secondoftwo 312, 785, 798, 828, 887, 891, 1094, 2646
\ltx@space 585, 587, 807, 816, 830, 833, 1205, 1211, 1378, 1425, 1621, 1649, 1692, 2563, 2570, 2617
\ltx@two 2743, 2755, 2773, 2784
\ltx@zero 477, 1619, 1672, 1772, 2108, 2613, 2626, 2644, 2746, 2751, 2776, 2781

M

\m@ne 1078
\M@sect 1979
\M@TitleReference 2359
\mag 1280, 2756, 2770, 2785, 2799
\mainmatter 60, 134
\makeatletter 11, 74, 101, 2141
\makeatother 16, 99
\makebox 169, 170
\MakeRobustcommand 233
\marginparsep 1321
\marginparwidth 1320
\meaning 2061
\mediaheight 1286
\mediawidth 1285
\MessageBreak 271, 514, 565, 571, 681, 1364, 1376, 1380, 1428, 1538, 1563, 1566, 1597, 1599, 1622, 1625, 1652, 1654, 1693, 1694, 2123, 2124, 2173, 2190, 2195, 2199, 2204, 2260, 2379, 2388, 2618, 2630, 2631

N

\NeedsTeXFormat 3, 188, 220, 965, 1003, 1026, 1053, 1107, 1147, 1238, 1256, 1437, 1608, 1701, 1719, 1832, 2076, 2479, 2491, 2542, 2885, 2940
\newcommand 18, 78, 85, 91, 167, 973, 980, 986, 1161, 1178, 1179, 1248, 1616, 2104, 2533, 2536, 2558, 2565, 2572, 2576, 2625, 2908
\newcount 2107
\newcounter 6, 911, 1015, 1731
\newdimen 2651, 2652

\newif 248, 701, 1844, 1857, 2092
\newlabel 2296, 2307, 2338
\newmarks 1625
\newpage 143
\nfss@text 933
\NR@temp 1962, 1963
\number 94, 109, 913, 925, 959, 1129, 1273, 1274, 1289, 1290, 1293, 1294, 1297, 1298, 1301, 1302, 1706, 1778, 1804, 1811, 2559, 2561, 2566, 2568, 2573, 2577, 2627, 2639, 2806, 2857, 2914
\numexpr 94, 109, 115, 927, 1131, 1192, 1706, 1778, 2253, 2264, 2298, 2383, 2392, 2574, 2578, 2770, 2799, 2921

O

\oddsidemargin 1313
\on@line 1708, 2188
\openin 2170

P

\PackageError 259, 270, 295, 305, 490, 1620
\PackageInfo 292, 521, 1709, 2177, 2189, 2280, 2322, 2343
\PackageWarning 346, 364, 381, 399, 616, 680, 1650, 1691, 2172
\PackageWarningNoLine 2259, 2378, 2387
\page 2420
\pageheight 1297, 1298, 2696
\pagestyle 49
\pagewidth 1301, 1302
\paperheight 1282
\paperwidth 1281
\pdf@escapehex 1499
\pdf@strcmp 477
\pdf@unescapehex 1500
\pdfhorigin 1307
\pdfpageattr 1464, 1470
\pdfpageheight 1306, 2683
\pdfpagesattr 1476, 1482
\pdfpagewidth 1305
\pdfvariable 1289, 1290, 1293, 1294, 1455, 1457
\pdfvorigin 1308
\ProcessOptions 217
\protect 777
\protected 239
\protected@write 716
\providecommand 223, 282, 1704, 2089, 2905, 2906, 2907
\ProvidesPackage 189, 221, 966, 1004, 1027, 1054, 1108, 1148, 1239, 1257, 1438, 1609, 1702, 1720, 1833, 2077, 2480, 2492, 2543, 2886, 2941

R

\read 2223
\ReadonlyShipoutCounter 1022
\refstepcounter 1044
\renewcommand 7, 46, 912
\RequirePackage . . . 191, 196, 224, 225, 226, 227, 228, 231, 267, 272, 280, 968, 1006, 1029, 1056, 1057, 1110, 1115, 1116, 1150, 1155, 1156, 1157, 1158, 1241, 1246, 1247, 1259, 1264, 1265, 1440, 1445, 1453, 1454, 1469, 1481, 1490, 1491, 1611, 1722, 1727, 1835, 1840, 1841, 2079, 2084, 2085, 2482, 2494, 2545, 2550, 2555, 2888, 2893, 2895, 2943
\reset@font 933
\rightarrow 45
\romannumeral 583, 805, 826, 1630, 2119, 2128, 2129

S

\section 63, 137, 145
\setcounter 919, 1018
\setlength 2917
\SetupKeyvalOptions 2093
\ShipoutBoxDepth 1334
\ShipoutBoxHeight 1333
\ShipoutBoxWidth 1332
\space 779, 1464, 1476, 2191, 2192, 2196, 2200, 2205
\spinemargin 1325
\stepcounter 19, 1020, 1733, 1734, 1743, 1763
\stockheight 1284, 2652
\stockwidth 1283, 2651, 2656, 2674, 2684, 2697

T

\tableofcontents 59, 132
\tex 2501, 2502, 2506
\textheight 1316
\textwidth 1315
\the 13, 153, 155, 430, 445, 461, 557, 562, 621, 627, 743, 757, 927, 1022, 1071, 1084, 1125, 1131, 1192, 1348, 1368, 1375, 1376, 1377, 1379, 1424, 1426, 1427, 1455, 1457, 1470, 1482, 1508, 1543, 1561, 1562, 1564, 1854, 2247, 2253, 2264, 2298, 2383, 2392
\thechapter 14
\thefoo 7, 12, 20
\theotype 2310
\thepage 43, 44, 45, 714, 718, 779, 941, 1802
\thezpage *16*, 1802, 1808
\thezref@unique *10*, 912, 1799, 1800, 1807, 1808, 1810, 2916, 2918, 2921, 2922, 2930
\title 2429, 2458
\toks@ 424, 430, 444, 445, 554, 557, 559, 562, 613, 620, 621, 627, 741, 743, 756, 757, 1067, 1071, 1079, 1084, 1364, 1368, 1374, 1375, 1536, 1543, 1560, 1561, 1848, 1854, 2232, 2247

\topmargin 1312
\TR@TitleReference . . 2354, 2414, 2453
\trimedge 1324
\trimtop 1327
\ttl@sect@i 2012
\ttl@straight@i 2018

U

\unexpanded 271, 276
\UniqueCounterCall 1176
\UniqueCounterNew 1159
\uppermargin 1328
\url 2449
\urluse 2444
\usepackage 9, 41, 48, 70, 72

V

\value 13, 1125, 1348, 1508
\verb 173
\voffset 1311

W

\write 705, 706, 1711

X

\x 331, 336, 724, 729, 898, 900, 1218, 1234, 1270, 1277, 1367, 1370, 1373, 1413, 1541, 1546, 1559, 1569, 1674, 1679, 1682, 1688, 1777, 1780, 2058, 2059, 2064, 2231, 2234, 2239, 2244, 2254, 2257, 2267, 2268, 2299, 2304
\XR@ext 2090

Y

\y 2060, 2061

Z

\z 2061, 2062, 2064
\z@ 2656, 2674, 2684, 2697, 2932
\Z@D@page 1179
\Z@L@LastPage 1069, 1081
\Z@L@main 1068, 1080
\Z@L@ZREF@temp 633, 637, 640, 651, 661
\zdotfill *19*, 170, 173, 2909
\zdotfillsetup *19*, 2908
\zexternaldocument *19*, 2135
\ziflastpage *12*, 1097
\zifrefundefined *8*, 763
\zkvlabel 980
\zlabel *11*, 83, 104, 138, 146, 973
\zlistpageattr 1503
\zlistpagelayout *15*, 1343
\zmakeperpage *16*, 1770
\znextpage *13*, 51, 54, 1175
\znextpagesetup *14*, 42, 1161
\znonextpagename 46, 1178, 1226
\zpageref *11*, 126, 995
\zposx *18*, 153, 2533, 2921, 2922
\zposy *18*, 155, 2533
\zref *11*, 25, 26, 27, 28, 112, 114, 123, 128, 129, 139, 986, 996
\ZREF@@@delprop . . . 435, 437, 472, 474
\ZREF@@@newprop 539, 543
\ZREF@@delprop 425, 434, 448, 465, 471, 481
\ZREF@@extract 809, 815
\ZREF@@makeperpage . . 1772, 1778, 1782
\ZREF@@newprop 517, 528, 531, 535
\ZREF@@perpage@step 1787, 1795
\ZREF@abspos@body@x@center 2836
\ZREF@abspos@body@x@left . 2834, 2861
\ZREF@abspos@body@x@right 2835, 2858
\ZREF@abspos@body@y@bottom 2841, 2853, 2875
\ZREF@abspos@body@y@center 2845, 2876
\ZREF@abspos@body@y@top 2837, 2842, 2846, 2874
\ZREF@abspos@foot@x@center 2851
\ZREF@abspos@foot@x@left 2849
\ZREF@abspos@foot@x@right 2850
\ZREF@abspos@foot@y@bottom 2852
\ZREF@abspos@head@x@center 2816, 2836, 2851
\ZREF@abspos@head@x@left 2802, 2813, 2817, 2834, 2849
\ZREF@abspos@head@x@right 2812, 2835, 2850
\ZREF@abspos@head@y@bottom 2826, 2838
\ZREF@abspos@head@y@center 2830
\ZREF@abspos@head@y@top 2820, 2827, 2831
\ZREF@abspos@marginpar@x@center 2870
\ZREF@abspos@marginpar@x@left 2856, 2867, 2871
\ZREF@abspos@marginpar@x@right 2866
\ZREF@abspos@marginpar@y@bottom 2875
\ZREF@abspos@marginpar@y@center 2876
\ZREF@abspos@marginpar@y@top . . 2874
\ZREF@abspos@media@height 2671, 2717, 2723
\ZREF@abspos@media@width 2653, 2710, 2714
\ZREF@abspos@media@x@center . . . 2712
\ZREF@abspos@media@x@left 2706, 2713
\ZREF@abspos@media@x@right 2709
\ZREF@abspos@media@y@bottom . . . 2719
\ZREF@abspos@media@y@center . . . 2722
\ZREF@abspos@media@y@top . 2716, 2734
\ZREF@abspos@origin@x . . . 2743, 2804
\ZREF@abspos@origin@y . . . 2773, 2822
\ZREF@abspos@paper@x@center 2731, 2879
\ZREF@abspos@paper@x@left 2725, 2803, 2877
\ZREF@abspos@paper@x@right 2728, 2878
\ZREF@abspos@paper@y@bottom 2735, 2881
\ZREF@abspos@paper@y@center 2739, 2882
\ZREF@abspos@paper@y@top 2734, 2736, 2740, 2821, 2880
\ZREF@abspos@stock@x@center . . . 2879
\ZREF@abspos@stock@x@left 2877
\ZREF@abspos@stock@x@right 2878

\ZREF@abspos@stock@y@bottom . . . 2881
\ZREF@abspos@stock@y@center . . . 2882
\ZREF@abspos@stock@y@top 2880
\ZREF@abspos@used 2603, 2605
\ZREF@absposnum 2574, 2578, 2580
\zref@absposnumused 2625
\zref@absposnumx . . . 2560, 2572, 2594
\zref@absposnumy . . . 2567, 2576, 2597
\zref@absposused 2602
\zref@absposx 2558, 2588
\zref@absposy 2565, 2591
\zref@addprop 6, 76, 360, 1023, 1037, 1041, 1118, 1121, 1275, 1291, 1295, 1299, 1303, 1456, 1458, 1471, 1483, 1658, 1843, 2488, 2556, 2557
\zref@addprops 6, 15, 341, 942, 1341, 1730, 2503
\ZREF@addtoks 755
\ZREF@base@ok 962, 969, 1007, 1030, 1058, 1111, 1151, 1242, 1260, 1441, 1612, 1723, 1836, 2080, 2483, 2495, 2546, 2889, 2944
\ZREF@call 1184, 1199, 1208, 1212, 1220
\ZREF@def@abspos 2588, 2591, 2594, 2597, 2599
\zref@def@absposnumx 2593
\ZREF@def@absposnumy 2599
\zref@def@absposnumy 2596
\zref@def@absposx 2587
\zref@def@absposy 2590
\ZREF@def@extract 846, 848
\zref@def@extract 8, 845
\ZREF@def@extractdefault . . . 857, 859
\zref@def@extractdefault 856
\ZREF@default 562, 563, 572
\zref@default . . . 8, 531, 807, 930, 932
\ZREF@delprop 413, 416, 418, 453, 456, 458
\zref@delprop 412, 452
\ZREF@df@dot 2903, 2907, 2936
\ZREF@df@min 2900, 2905, 2923
\ZREF@df@unit 2897, 2906, 2917
\ZREF@dotfill 2919, 2926, 2935
\ZREF@ENV@line 2954, 2957
\zref@env@line 2949, 2950
\ZREF@extract 804, 821, 824, 876
\zref@extract 7, 95, 96, 109, 140, 804, 824, 853, 871, 876, 993, 1134, 1230, 1379, 1426, 1427, 1550, 1587, 1807, 1808, 1901, 2534, 2537, 2710, 2714, 2717, 2723, 2729, 2732, 2737, 2741, 2762, 2791, 2805, 2807, 2809, 2814, 2818, 2823, 2824, 2828, 2832, 2839, 2843, 2847, 2854, 2859, 2862, 2863, 2868, 2872
\ZREF@extractdefault 825, 841, 844, 875
\zref@extractdefault . 7, 116, 117, 817, 844, 864, 870, 875, 1090, 1091, 1188, 1203, 1249, 1810, 2352, 2355, 2356, 2360, 2361, 2364, 2366, 2367, 2369, 2371, 2561, 2568, 2611, 2639, 2806, 2857
\ZREF@false 677, 687
\ZREF@foundfalse 2413
\ZREF@foundtrue 2460
\ZREF@getcurrent . . 582, 593, 596, 874
\zref@getcurrent 7, 596, 869, 874
\zref@hex 1455, 1457, 1470, 1482, 1496, 1499
\zref@ifabsposnumundefined 2638, 2643
\zref@ifabsposundefined 2636
\ZREF@IfDefinable 242, 763, 995, 998, 1097, 1139, 1175, 1343, 1503, 1770, 1824, 1891, 1894, 2135, 2518, 2523, 2528, 2909
\ZREF@iflastpage . . . 1098, 1100, 1100
\zref@iflastpage 12, 1089, 1103
\zref@iflistcontainsprop 6, 310, 345, 363, 380, 398, 648, 658
\zref@iflistundefined 6, 289, 300, 304, 311
\zref@ifpropundefined 7, 485, 489, 519, 548, 615, 828, 1422, 1671, 2407
\ZREF@ifrefcontainsprop . . . 787, 795
\zref@ifrefcontainsprop 8, 783, 1424, 2368, 2464, 2465
\ZREF@ifrefundefined 765, 768, 1185, 1196, 1206
\zref@ifrefundefined 8, 760, 770, 776, 784, 827, 1197, 1377, 1564, 1800, 2607, 2637, 2645, 2918
\ZREF@immediatefalse 678
\ZREF@immediatetrue 675, 704
\ZREF@label 603, 627, 637, 640, 710, 1071, 1084
\zref@label 7, 597, 977
\zref@labelbykv 630, 984
\zref@labelbylist 7, 598, 600, 1125, 1799, 2520
\zref@labelbyprops 7, 88, 610, 1183, 2525, 2530, 2916, 2930
\zref@listexists 6, 303, 322, 342, 361, 377, 396, 419, 459, 602
\zref@listforloop 321, 657
\zref@listpageattr 1503
\zref@listpagelayout 1343
\zref@localaddprop 395
\zref@localaddprops 376
\zref@localdelprop 415, 455, 669
\ZREF@mainlist 598, 936, 939, 942, 1023, 1037, 1041, 1843, 2488
\ZREF@makeperpage@opt . . . 1772, 1775
\ZREF@MARKS@DefineProp 1634, 1635, 1636, 1670
\zref@marks@register 1616, 1621, 1653, 1692
\ZREF@name 229, 259, 270, 292, 295, 305, 346, 364, 381, 399, 490, 509, 521, 525, 549, 564, 616, 680, 692, 1620
\ZREF@NAME@bot 1649, 1669
\ZREF@NAME@first 1648, 1668

\ZREF@NAME@top 1647, 1667
\zref@newlabel 7, 282, 285, 750, 2251, 2337
\zref@newlist 6, 288, 939, 1062, 1117, 1644, 1729, 2499
\ZREF@newprop 498, 501, 504
\zref@newprop 6, 12, 13, 14, 75, 495, 940, 941, 944, 951, 955, 959, 1022, 1036, 1040, 1272, 1288, 1292, 1296, 1300, 1332, 1333, 1334, 1337, 1338, 1339, 1455, 1457, 1470, 1482, 1675, 1683, 2086, 2087, 2088, 2408, 2501, 2502, 2948, 2949
\ZREF@NewPropAnchor . . 943, 2137, 2487
\ZREF@NewPropPageValue 958, 1119, 1728
\ZREF@NewPropTheotype 954, 2308
\ZREF@NewPropTitle . . . 949, 1842, 2138
\ZREF@nextpage 1176, 1180
\ZREF@nil 545, 797, 836, 2224, 2230, 2236, 2241, 2251, 2267, 2296, 2304, 2395, 2402, 2411, 2414, 2453
\ZREF@NOVALUE 803
\ZREF@novalue 796, 797, 803
\ZREF@np@call@next . . 1170, 1174, 1229
\ZREF@np@call@nonext 1167, 1173, 1225
\ZREF@np@call@unknown 1163, 1172, 1221
\ZREF@np@setup@i 1162, 1165
\ZREF@np@setup@ii 1166, 1169
\ZREF@number 924, 1550, 1556, 1618, 2644
\ZREF@org@@begintheorem 2068
\ZREF@org@@caption 1908
\ZREF@org@@chapter 1920, 1976
\ZREF@org@@opargbegintheorem . . 2053
\ZREF@org@@part 1914
\ZREF@org@@schapter 1938
\ZREF@org@@sect 1926
\ZREF@org@@spart 1932
\ZREF@org@@ssect 1944
\ZREF@org@@stpelt . . 1759, 1764, 1768
\ZREF@org@beamer@section 1994
\ZREF@org@beamer@subsection . . . 2000
\ZREF@org@beamer@subsubsection 2006
\ZREF@org@descriptionlabel 1950
\ZREF@org@lst@MakeCaption 2046
\ZREF@org@LT@c@ption 2032
\ZREF@org@M@sect 1985
\ZREF@org@refstepcounter 1046
\ZREF@org@stepcounter 1733, 1738, 1763
\ZREF@org@testdef 1350, 1352, 1510, 1512
\ZREF@org@thepage 714, 718
\ZREF@org@ttl@sect@i 2014
\ZREF@org@ttl@straight@i 2025
\ZREF@org@write 705, 706
\ZREF@P 505, 506, 508, 510, 519, 522, 526, 536, 537, 539, 540, 541, 545, 721, 725, 726, 735, 739, 744, 745
\ZREF@pa@AfterLastShipout 1506, 1604
\ZREF@pa@AtVeryEnd . . 1519, 1522, 1593
\ZREF@pa@ListPage 1540, 1558
\ZREF@pa@listtrue 1504
\ZREF@page@max . 1348, 1414, 1508, 1570
\zref@pageattr 1548
\zref@pageattr@used 1555
\ZREF@pagenum@last 1202, 1205
\ZREF@pagenum@this 1187, 1192, 1195, 1205, 1211
\ZREF@par 508, 533
\ZREF@param 422, 423, 442, 460, 477, 646, 647, 648, 652, 673, 674, 677, 682
\ZREF@patch . 249, 1043, 1905, 1911, 1917, 1923, 1929, 1935, 1941, 1947, 1978, 1991, 1997, 2003, 2011, 2017, 2030, 2038, 2050, 2065
\zref@pdfpageattr 1466, 1548, 1554, 1565
\zref@pdfpageattr@used 1467
\zref@pdfpagesattr . . 1478, 1585, 1598
\zref@pdfpagesattr@used . 1479, 1590
\ZREF@pl@AfterLastShipout 1346, 1433
\ZREF@pl@AtVeryEnd 1359, 1362
\ZREF@pl@ListEntry 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1421
\ZREF@pl@ListPage 1366, 1372
\ZREF@pl@listtrue 1344
\zref@pos@label@used 2617
\zref@pos@num@used 2631
\zref@prop 324, 332, 333, 337, 658, 662
\zref@propexists 7, 344, 362, 379, 397, 488, 578, 647, 668, 987
\ZREF@refname@next 1190, 1197, 1206, 1230
\ZREF@refname@this 1182, 1183, 1185, 1188
\ZREF@RefPrefix . 284, 286, 1354, 1514
\ZREF@refused 773, 775
\zref@refused 8, 769, 772, 849, 860, 992, 999, 1101, 1102, 1137, 1252, 1556, 1591, 1899, 2606, 2614, 2627
\zref@require@unique 10, 909, 1732, 2894
\ZREF@Robust 232, 238, 244, 285, 288, 303, 310, 341, 360, 376, 395, 412, 415, 452, 455, 488, 495, 547, 577, 597, 600, 610, 630, 702, 772, 845, 856, 867, 883, 909, 929, 935, 1136, 1555, 1590, 1846, 1859, 2587, 2590, 2593, 2596, 2602
\ZREF@SavedEscapechar 461, 468
\zref@savepos . . . 18, 2504, 2512, 2514
\ZREF@savepos@ok 2539, 2551
\zref@setcurrent 6, 81, 541, 577, 697, 1045
\zref@setdefault 8, 929, 932

\zref@setmainlist *9*, 935
\zref@showprop 547
\ZREF@STAR 1642, 1666
\ZREF@stripperiod 1870, 1878
\ZREF@temp 193, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 233, 234, 440, 441, 442, 720, 741, 742, 750, 1266, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1305, 1306, 1307, 1308, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1353, 1354, 1446, 1459, 1472, 1484, 1487, 1492, 1493, 1494, 1495, 1513, 1514, 1522, 1523, 1524, 1525, 1578, 1579, 1580, 1581, 1641, 1642, 2019, 2020, 2111, 2121, 2124, 2128, 2610, 2613, 2614, 2743, 2746, 2751, 2755, 2757, 2760, 2773, 2776, 2781, 2784, 2786, 2789, 2914
\ZREF@TempName . . 1617, 1629, 1630, 1632, 1658, 1671, 1675, 1683, 1694
\ZREF@TempNum 1618, 1619, 1623, 1630, 1672, 1685
\zref@thepage *13*, 1133, 1142
\zref@thepage@atbegshi@hook 1120, 1124
\zref@thepage@name *13*, 1128, 1134, 1137, 1191
\zref@thepage@refused . . . 1136, 1141
\ZREF@titleref 1895, 1897
\zref@titleref@cleanup . . 1846, 1886
\zref@titleref@current 949, 1865, 1869, 1870, 1889
\ZREF@titleref@hook 1845, 1849, 1853, 1876
\zref@titleref@setcurrent 1859, 1907, 1913, 1919, 1925, 1931, 1937, 1943, 1949, 1957, 1960, 1964, 1968, 1970, 1981, 1983, 1993, 1999, 2005, 2013, 2021, 2023, 2033, 2042, 2052, 2067
\zref@titleref@stripperiodtrue 1858
\ZREF@true 674, 688
\ZREF@u@getcurrent 591
\zref@unhex 1497, 1500, 1549, 1586
\ZREF@UpdatePdfTeX 247
\ZREF@value 557, 558, 571
\ZREF@wrapper@babel 900, 906
\zref@wrapper@babel *10*, 140, 765, 773, 846, 857, 883, 977, 984, 988, 1098, 1895, 2588, 2591, 2594, 2597, 2603
\zref@wrapper@immediate *10*, 87, 636, 702, 1083
\ZREF@wrapper@unexpanded . . . 867, 881
\zref@wrapper@unexpanded *10*, 868, 873, 878, 2347
\ZREF@wu@extract 819, 871
\ZREF@wu@extractdefault . . . 839, 870
\ZREF@wu@getcurrent 591, 869
\ZREF@X 497, 500, 537
\zref@xr@ 2101
\ZREF@xr@@AddUrl 2113, 2116
\ZREF@xr@@input 2244, 2339
\ZREF@xr@AddURL 2109, 2162, 2441
\ZREF@xr@checkfile . . 2166, 2169, 2219
\ZREF@xr@checkkey 2397, 2406
\ZREF@xr@checklist 2267, 2395
\zref@xr@ext *20*, 2089, 2157
\ZREF@xr@externaldocument 2144, 2147, 2151
\ZREF@xr@externalfile 2154, 2155, 2273, 2315
\ZREF@xr@file . . . 2155, 2170, 2173, 2179, 2190, 2213, 2261, 2380, 2389
\ZREF@xr@filelist 2153, 2211, 2214, 2216, 2217, 2245, 2246
\ZREF@xr@found . 2181, 2191, 2253, 2298
\ZREF@xr@graburl 2157, 2159
\ZREF@xr@ignored@empty 2182, 2194, 2196, 2263, 2264
\ZREF@xr@ignored@ltx 2184, 2203, 2205, 2391, 2392
\ZREF@xr@ignored@zref 2183, 2198, 2200, 2382, 2383
\ZREF@xr@line . . 2223, 2224, 2236, 2241
\ZREF@xr@list 2257, 2258
\ZREF@xr@ltx@ignorewarning 2386
\ZREF@xr@newlabel 2239, 2338
\ZREF@xr@prefix 2152, 2252, 2288, 2292, 2297, 2323, 2330, 2334
\ZREF@xr@process@label . . 2241, 2296
\ZREF@xr@process@zreflabel 2236, 2251
\ZREF@xr@processfile 2169, 2222
\ZREF@xr@processline 2224, 2230
\ZREF@xr@refname 2252, 2278, 2285, 2297, 2320, 2327
\ZREF@xr@relax 2340, 2427
\ZREF@xr@scanparams 2302, 2411
\ZREF@xr@scantitleref . . . 2414, 2453
\ZREF@xr@temp 2426, 2427
\ZREF@xr@tempname 2255, 2256, 2276, 2281, 2292, 2300, 2301, 2318, 2334
\ZREF@xr@temprefname 2256, 2268, 2270, 2286, 2301, 2303, 2310, 2313, 2328
\ZREF@xr@theURL 2119, 2121, 2123, 2129, 2164, 2445
\ZREF@xr@tolabel . . . 2292, 2334, 2341
\ZREF@xr@URL . . . 2107, 2117, 2118, 2119
\ZREF@xr@url . . . 2160, 2162, 2163, 2471
\ZREF@xr@urlcheck . . 2276, 2318, 2463
\ZREF@xr@zref@ignorewarning 2288, 2330, 2377
\ZREF@xr@zref@newlabel . . 2234, 2337
\ZREF@xr@zreflabelfalse 2143
\ZREF@xr@zreflabeltrue 2146
\ZREF@zref 988, 991
\ZREF@zsavepos . 2509, 2520, 2525, 2530

\zrefused *11*, 92, 93, 161, 162, 163, 998
\zruns *16*, 1704
\zsavepos *17*, 157, 158, 2518
\zsaveposx *17*, 2523
\zsaveposy 2528
\zthepage *13*, 1139
\ztitleref *17*, 1894
\ztitlerefsetup *17*, 1879
\ztotpages *15*, 124, 1248
\zunknownnextpagename . *14*, 1179, 1222
\zunmakeperpage *16*, 1824
\zxrsetup *19*, 2104