

The zref package

Heiko Oberdiek*

2026-03-07 v2.37

Abstract

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

Contents

1	Introduction	4
1.1	Standard \LaTeX behaviour	4
1.2	Basic idea	4
1.3	Interfaces	5
2	Interface for programmers	5
2.1	Entities	5
2.2	Property list	6
2.3	Property	6
2.4	Reference generation	7
2.5	Data extraction	7
2.6	Setup	8
2.7	Declared properties	9
2.8	Wrapper for advanced situations	10
2.9	Counter for unique names	10
3	User interface	10
3.1	Module <code>user</code>	10
3.2	Module <code>abspage</code>	11
3.3	Module <code>lastpage</code>	12
3.3.1	Tests for last page	12
3.3.2	Example	12
3.4	Module <code>thepage</code>	13
3.5	Module <code>nextpage</code>	13
3.5.1	Configuration	14
3.5.2	Example	14
3.6	Module <code>totpages</code>	15
3.7	Module <code>pagelayout</code>	15
3.8	Module <code>marks</code>	16
3.9	Module <code>runs</code>	16
3.10	Module <code>perpage</code>	16
3.11	Module <code>counter</code>	17
3.12	Module <code>titleref</code>	17
3.13	Module <code>savepos</code>	17
3.14	Module <code>abspos</code>	18

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

3.15	Module <code>dotfill</code>	19
3.16	Module <code>env</code>	19
3.17	Module <code>xr</code>	19
3.18	Module <code>pageattr</code>	20
4	ToDo	20
5	Example	20
6	Implementation	23
6.1	Package <code>zref</code>	23
6.1.1	Identification	23
6.1.2	Load basic module	23
6.1.3	Process options	23
6.2	Module <code>base</code>	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 <code>babel</code>	38
6.2.11	Unique counter support	39
6.2.12	Utilities	39
6.2.13	Setup	39
6.3	Module <code>user</code>	40
6.4	Module <code>abspage</code>	42
6.5	Module <code>counter</code>	42
6.6	Module <code>lastpage</code>	43
6.7	Module <code>thepage</code>	44
6.8	Module <code>nextpage</code>	45
6.9	Module <code>totpages</code>	46
6.10	Module <code>pagelayout</code>	47
6.10.1	Define layout properties	47
6.11	Module <code>pageattr</code>	50
6.12	Module <code>marks</code>	53
6.13	Module <code>runs</code>	55
6.14	Module <code>perpage</code>	55
6.15	Module <code>titleref</code>	57
6.15.1	Implementation	58
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 <code>memoir</code>	60
6.15.6	Class <code>beamer</code>	61
6.15.7	Package <code>titlesec</code>	61
6.15.8	Package <code>longtable</code>	62
6.15.9	Package <code>listings</code>	62
6.15.10	Theorems	62
6.16	Module <code>xr</code>	63
6.17	Module <code>hyperref</code>	70
6.18	Module <code>savepos</code>	70
6.18.1	Identification	70
6.18.2	Availability	71
6.18.3	Setup	71

6.18.4	User macros	71
6.19	Module <code>abspos</code>	72
6.19.1	Identification	72
6.19.2	Media	74
6.19.3	Paper	76
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 <code>dotfill</code>	80
6.21	Module <code>env</code>	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
[2026-03-07 v2.37]	89

10 Index **89**

1 Introduction

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

Unhappily L^AT_EX does not provide an interface for adding another properties. Packages such as `hyperref`, `nameref`, or `titleref` are forced to use ugly hacks to extend the reference system. These ugly hacks are one of the causes for `hyperref`’s difficulty regarding compatibility with other packages.

1.1 Standard L^AT_EX behaviour

References are created by the `\label` command:

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

Now L^AT_EX records the section number 2.1 and the page 7 in the reference. Internally the reference is a list with two entries:

```
\r@myref → {2.1}{7}
```

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

There are several tries to add new properties:

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

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

1.2 Basic idea

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

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

The entries have the following format:

```
\xref_{\langle your key \rangle}_key{\langle some text \rangle}
```

This approach is much more flexible:

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

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

1.3 Interfaces

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

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

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

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

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

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

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

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

2 Interface for programmers

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

2.1 Entities

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

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

The generic format of a entry is:

```
\Z@R@(\refname) → \(\propname){\langle value \rangle}
```

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

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

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

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

2.2 Property list

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

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

Declares a new empty property list.

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

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

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

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

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

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

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

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

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

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

2.3 Property

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

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

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

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

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

This sets the current value of the property `<propname>`. It is a generalization of

setting L^AT_EX's `\currentlabel`.

```
\zref@getcurrentexp2 {⟨propname⟩}
```

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

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

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

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

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

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

2.4 Reference generation

```
\zref@label {⟨refname⟩}
```

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

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

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

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

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

```
\zref@newlabel {⟨refname⟩} {...}
```

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

2.5 Data extraction

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

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

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

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

Example for page references:

```

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

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

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

```
\zref@refused {<refname>}
```

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

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

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

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

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

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

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

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

Test whether a reference provides a property.

2.6 Setup

```
\zref@default
```

Holds the global default for unknown values.

```
\zref@setdefault {<value>}
```

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

`\zref@setmainlist {(value)}`

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

2.7 Declared properties

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

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

2.8 Wrapper for advanced situations

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

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

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

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

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

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

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

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

2.9 Counter for unique names

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

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

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

```
\thezref@unique
```

This command is used to generate unique label names.

3 User interface

3.1 Module user

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

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

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

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

`\zlabel {⟨refname⟩}babel`

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

`\zref [⟨propname⟩] {⟨refname⟩}babel`

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

`\zref{x} ≡ \zref[default]{x}`

`\zpageref {⟨refname⟩}babel`

Convenience macro, similar to `\pageref`.

`\zpageref{x} ≡ \zref[page]{x}`

`\zrefused {⟨refname⟩}babel`

Some of the user commands in the modules are expandable. The use of such commands do not cause any undefined reference warnings, because inside of expandable contexts this is not possible. However, if there is a place outside of expandable contexts, `\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 L^AT_EX count `\ReadonlyShipoutCounter` to keep track of the page numbers.

Thus you can reference the absolute page number:

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

The example also makes use of module `lastpage`.

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

3.3 Module `lastpage`

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

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

or

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

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

3.3.1 Tests for last page

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

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

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

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

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

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

3.3.2 Example

```
1 <*example-lastpage>
2 %<<END_EXAMPLE
3 \NeedsTeXFormat{LaTeX2e}
4 \documentclass{report}
5
6 \newcounter{foo}
7 \renewcommand*{\thefoo}{\Alph{foo}}
8
9 \usepackage{zref-lastpage,zref-user}[2019/11/29]
10
11 \makeatletter
12 \zref@newprop{thefoo}{\thefoo}
13 \zref@newprop{valuefoo}{\the\value{foo}}
14 \zref@newprop{chapter}{\thechapter}
15 \zref@addprops{LastPage}{thefoo,valuefoo,chapter}
16 \makeatother
17
18 \newcommand*{\foo}{%
```

```

19 \stepcounter{foo}%
20 [Current foo: \thefoo]%
21 }
22
23 \begin{document}
24 \chapter{First chapter}
25 Last page is \zref{LastPage}.\
26 Last chapter is \zref[chapter]{LastPage}.\
27 Last foo is \zref[thefoo]{LastPage}.\
28 Last value of foo is \zref[valuefoo]{LastPage}.\
29 \foo
30 \chapter{Second chapter}
31 \foo\foo\foo
32 \chapter{Last chapter}
33 \foo
34 \end{document}
35 %END_EXAMPLE
36 \example-lastpage

```

3.4 Module `thepage`

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

`\zthepage {⟨absolute page number⟩}`

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

`\zref@thepage@nameexp {⟨absolute page number⟩}`

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

`\zref@thepageexp {⟨absolute page number⟩}`
`\zref@thepage@refused {⟨absolute page number⟩}`

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

3.5 Module `nextpage`

`\znextpage`

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

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

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

3.5.1 Configuration

The behaviour can be configured by the following macros.

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

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

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

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

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

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

3.5.2 Example

```

37 <*example-nextpage>
38 %<<END_EXAMPLE
39 \documentclass{book}
40
41 \usepackage{zref-nextpage}[2019/11/29]
42 \znextpagesetup
43   {\thepage}% next page is unknown
44   {\thepage\ (#1)}% this page is last page
45   {\thepage\ $\rightarrow$ #1}% next page is known
46 \renewcommand*{\znonextpagename}{last page}
47
48 \usepackage{fancyhdr}
49 \pagestyle{fancy}
50 \fancyhf{}
51 \fancyhead[LE,RO]{\znextpage}
52 \fancypagestyle{plain}{%
53   \fancyhf{}%
54   \fancyhead[LE,RO]{\znextpage}%
55 }
56
57 \begin{document}
58 \frontmatter
59 \tableofcontents
60 \mainmatter
61 \chapter{Hello World}
62 \clearpage
63 \section{Last section}
64 \end{document}
65 %END_EXAMPLE
66 </example-nextpage>

```

3.6 Module `totpages`

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

`\ztotpagesexp`

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

3.7 Module `pagelayout`

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

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

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

List of properties:

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

`\zlistpagelayout`

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

3.8 Module marks

ToDo.

3.9 Module runs

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

`\zrunsexp`

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

3.10 Module perpage

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

`\zmakeperpage [reset] {counter}`

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

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

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

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

`\thezpage`
counter `zpage`

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

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

`\zunmakeperpage {counter}`

The reset mechanism for this counter is deactivated.

3.11 Module counter

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

3.12 Module titleref

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

```
\ztitleref {<refname>}babel
```

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

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

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

`title=<value>`
Sets the current title.

`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 `XgTeX`). 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.

$\backslash\text{zposx}^{\text{exp}} \{ \langle \text{refname} \rangle \}$ $\backslash\text{zposy}^{\text{exp}} \{ \langle \text{refname} \rangle \}$
--

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

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

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

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

<code>\zref@savepos</code>

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{ \langle \text{refname} \rangle }{ \text{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{ \langle \text{label} \rangle }{ \langle \text{value} \rangle }{ \langle \text{position} \rangle }
\zref@absposy{ \langle \text{label} \rangle }{ \langle \text{value} \rangle }{ \langle \text{position} \rangle }

```

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.

`\langle \text{label} \rangle` is a label set with `\zlabel` or `\zsavepos` that allows to retrieve the absolute page number.

`\langle \text{position} \rangle` 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 `\langle \text{value} \rangle` can be seen in the following table. Be aware that the code makes some assumptions which are perhaps not always true – for example that the left of the head is identical to the left of the body.

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

3.15 Module `dotfill`

`\zdotfill`

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

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

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

`min=<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>] babel {<external document>} [<url>]`

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

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

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

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

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

The following setup options are available:

ext: It sets the default extension.

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

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

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

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

`\zref@xr@ext`

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

3.18 Module `pageattr`

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

```
\zref@pdfpageattr{<absolute page number>}
\zref@pdfpagesattr{<absolute page number>}
```

4 ToDo

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

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

5 Example

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

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

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

```
74 \makeatletter
75 \zref@newprop{chaptitle}{}
76 \zref@addprop{main}{chaptitle}
77
78 \newcommand*{\ChapterStart}[2]{%
```

```

79 \cleardoublepage
80 \def\current@chapid{#1}%
81 \zref@setcurrent{chaptitle}{#2}%
82 \chapter{#2}%
83 \zlabel{chap:#1}%
84 }
85 \newcommand*\ChapterStop}{%
86 \cleardoublepage
87 \zref@wrapper@immediate{%
88 \zref@labelbyprops{chapend:\current@chapid}{abspage}%
89 }%
90 }

```

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

```

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

```

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

```

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

```

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 is stored in references and are available for calculations in the next \LaTeX compile run.

```

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

```

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

```

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

```

Test for module `\dotfill`.

```

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

```

186 `</example>`

6 Implementation

6.1 Package zref

6.1.1 Identification

```
187 <*package>
188 \NeedsTeXFormat{LaTeX2e}
189 \ProvidesPackage{zref}
190 [2026-03-07 v2.37 A new reference scheme for LaTeX (HO)]%
```

6.1.2 Load basic module

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

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

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

6.1.3 Process options

Known modules are loaded and the release date is checked.

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

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:

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

`\ZREF@`: Internal macros.

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

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

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

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

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

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

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

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

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

`ZREF@TR`: Setup for module `titleref`.

6.2.2 Identification

```
219  $\langle *base \rangle$ 
220 \NeedsTeXFormat{LaTeX2e}
221 \ProvidesPackage{zref-base}%
222 [2026-03-07 v2.37 Module base for zref (H0)]%
```

6.2.3 Utilities

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

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

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

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

`\ZREF@Robust`

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

237 }%
```

`\ZREF@Robust`

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

241 }
```

`\ZREF@ifDefinable`

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

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

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

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

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

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

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

6.2.4 Check for ε -TeX

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

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

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

6.2.5 Auxiliary file stuff

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

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

`\ZREF@RefPrefix`

```
284 \def\ZREF@RefPrefix{ZOR}
```

`\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:

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

```

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

```

6.2.6 Property lists

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

`\zref@listforloop`

```

321 \def\zref@listforloop#1#2{%
322 \zref@listexists{#1}{%
323 \expandafter\expandafter\expandafter\@tfor
324 \expandafter\expandafter\expandafter\zref@prop
325 \expandafter\expandafter\expandafter:%
326 \expandafter\expandafter\expandafter=%
327 \csname Z@L@#1\endcsname
328 \do{%

```

```

329     \begingroup
330     \escapechar=-1 %
331     \edef\x{\endgroup
332     \def\noexpand\zref@prop{%
333     \expandafter\string\zref@prop
334     }%
335     }%
336     \x
337     #2\zref@prop
338     }%
339 }%
340 }

```

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

```

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

```

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

```

360 \ZREF@Robust\def\zref@addprop#1#2{%
361 \zref@listexists{#1}{%
362 \zref@propexists{#2}{%
363 \zref@iflistcontainsprop{#1}{#2}{%
364 \PackageWarning\ZREF@name{%
365 Property ‘#2’ is already in list ‘#1’%
366 }%
367 }{%
368 \begingroup\expandafter\endgroup
369 \expandafter\g@addto@macro
370 \csname Z@L@#1\expandafter\endcsname
371 \expandafter{\csname#2\endcsname}%
372 }%
373 }%
374 }%
375 }

```

`\zref@localaddprops`

```

376 \ZREF@Robust\def\zref@localaddprops#1#2{%
377 \zref@listexists{#1}{%
378 \comma@parse{#2}{%
379 \zref@propexists\comma@entry{%
380 \zref@iflistcontainsprop{#1}\comma@entry{%
381 \PackageWarning\ZREF@name{%
382 Property ‘\comma@entry’ is already in list ‘#1’%

```

```

383     }%
384   }{%
385     \begingroup\expandafter\endgroup
386     \expandafter\ltx@LocalAppendToMacro
387     \csname Z@L@#1\expandafter\endcsname
388     \expandafter{\csname\comma@entry\endcsname}%
389   }%
390 }%
391 \ltx@gobble
392 }%
393 }%
394 }

```

\zref@localaddprop

```

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

```

```

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

```

\zref@delprop

```

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

```

\zref@localdelprop

```

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

```

\ZREF@delprop

```

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

```

\ZREF@@delprop

```

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

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

451 }{%

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

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

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

\ZREF@@delprop Caution: #1 might be an \if or similar token.
471 \def\ZREF@@delprop#1{%
472   \expandafter\ZREF@@delprop\expandafter{\string#1}#1%
473 }%

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

```

484 }

6.2.7 Properties

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

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

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

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

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

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

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

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

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

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

`\ZREF@newprop`

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

```

522     New property: \ZREF@P
523   }%
524 }{%
525   \@PackageError\ZREF@name{%
526     Property ‘\ZREF@P’ already exists%
527   }\@ehc
528   \def\ZREF@@newprop[#1]##2{\endgroup}%
529 }%
530 \fi
531 \ifnextchar[\ZREF@@newprop{\ZREF@@newprop[\zref@default]}%
532 }

```

\ZREF@par

```

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

```

\ZREF@@newprop

```

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

```

\zref@showprop

```

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

```

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

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

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

```
582 \def\ZREF@getcurrent#1{%
583   \romannumeral0%
584   \ltx@ifundefined{Z@C@#1}{%
585     \ltx@space
586   }{%
587     \expandafter\expandafter\expandafter\ltx@space
588     \csname Z@C@#1\endcsname
589   }%
590 }
```

`\ZREF@u@getcurrent`

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

`\zref@getcurrent`

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

6.2.8 Reference generation

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

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

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

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

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

```
610 \ZREF@Robust\def\zref@labelbyprops#1#2{%
611   \@bsphack
612   \begingroup
613   \toks@{%
614     \comma@parse{#2}{%
615       \zref@ifpropundefined\comma@entry{%
616         \PackageWarning\ZREF@name{%
617           Property ‘\comma@entry’ is not known%
618         }%
619       }{%
620         \toks@\expandafter{%
621           \the\expandafter\toks@\csname\comma@entry\endcsname
622         }%

```

```

623     }%
624     \ltx@gobble
625     }%
626     \expandafter\endgroup
627     \expandafter\ZREF@label\expandafter{\the\toks@}{#1}%
628     \@esphack
629 }

```

\zref@labelbykv

```

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

645 \kv@define@key{ZREF@LABEL}{prop}{-%
646   \edef\ZREF@param{#1}%
647   \zref@propexists\ZREF@param{%
648     \zref@iflistcontainsprop{ZREF@temp}\ZREF@param}{-%
649     \begingroup\expandafter\endgroup
650     \expandafter\ltx@LocalAppendToMacro
651     \expandafter\Z@L@ZREF@temp
652     \expandafter{\csname\ZREF@param\endcsname}%
653   }%
654 }%
655 }

656 \kv@define@key{ZREF@LABEL}{list}{-%
657   \zref@listforloop{#1}{-%
658     \zref@iflistcontainsprop{ZREF@temp}\zref@prop}{-%
659     \begingroup\expandafter\endgroup
660     \expandafter\ltx@LocalAppendToMacro
661     \expandafter\Z@L@ZREF@temp
662     \expandafter{\csname\zref@prop\endcsname}%
663   }%
664   \ltx@gobble
665 }%
666 }

667 \kv@define@key{ZREF@LABEL}{delprop}{-%
668   \zref@propexists{#1}{-%
669     \zref@localdelprop{ZREF@temp}{#1}%
670   }%
671 }

672 \kv@define@key{ZREF@LABEL}{immediate}[true]{-%
673   \edef\ZREF@param{#1}%
674   \ifx\ZREF@param\ZREF@true
675     \ZREF@immediatetrue
676   \else
677     \ifx\ZREF@param\ZREF@false
678       \ZREF@immediatefalse
679     \else
680       \PackageWarning\ZREF@name{%
681         Option ‘immediate’ expects ‘true’ or ‘false’.\MessageBreak
682         Ignoring invalid value ‘\ZREF@param’%

```

```

683     }%
684     \fi
685     \fi
686 }

\ZREF@false
687 \def\ZREF@false{false}

\ZREF@true
688 \def\ZREF@true{true}

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

```

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

```
701 \newif\ifZREF@immediate
```

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

```

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

```

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

```

710 \def\ZREF@label#1#2{%
711   \if@filesw
712     \begingroup
713       \ifZREF@immediate
714         \let\ZREF@org@thepage\thepage
715       \fi
716       \protected@write\@auxout{%
717         \ifZREF@immediate
718           \let\thepage\ZREF@org@thepage
719         \fi
720       \let\ZREF@temp\ltx@empty
721       \@tfor\ZREF@P:=#1\do{%
722         \begingroup
723           \escapechar=-1 %
724           \edef\x{\endgroup
725             \def\noexpand\ZREF@P{%
726               \expandafter\string\ZREF@P

```

```

727         }%
728     }%
729     \x
730     \expandafter\ifx
731         \csname
732             \ifZREF@immediate
733                 relax%
734             \else
735                 Z@X@\ZREF@P%
736             \fi
737         \endcsname
738         \noexpand
739         \expandafter\let\csname Z@C@\ZREF@P\endcsname\relax
740     \fi
741     \toks@\expandafter{\ZREF@temp}%
742     \edef\ZREF@temp{%
743         \the\toks@
744         \ltx@backslashchar\ZREF@P{%
745         \expandafter\noexpand\csname Z@C@\ZREF@P\endcsname
746     }%
747 }%
748 }%
749 }{%
750     \string\zref@newlabel{#2}{\ZREF@temp}%
751 }%
752 \endgroup
753 \fi
754 }
755 \def\ZREF@addtoks#1{%
756     \toks@\expandafter\expandafter\expandafter{%
757         \expandafter\the\expandafter\toks@#1%
758     }%
759 }

```

6.2.9 Reference querying and extracting

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

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

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

```

760 \def\zref@ifrefundefined#1{%
761     \ltx@ifundefined{Z@R@#1}%
762 }

```

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

```

763 \ZREF@IfDefinable\zifrefundefined\def{%
764     #1{%
765         \zref@wrapper@babel\ZREF@ifrefundefined{#1}%
766     }%
767 }

```

`\ZREF@ifrefundefined`

```
768 \def\ZREF@ifrefundefined#1{%
769   \zref@refused{#1}%
770   \zref@ifrefundefined{#1}%
771 }
```

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

```
772 \ZREF@Robust\def\zref@refused#1{%
773   \zref@wrapper@babel\ZREF@refused{#1}%
774 }
```

`\ZREF@refused`

```
775 \def\ZREF@refused#1{%
776   \zref@ifrefundefined{#1}{%
777     \protect\G@refundefinedtrue
778     \@latex@warning{%
779       Reference ‘#1’ on page \thepage \space undefined%
780     }%
781   }{%}
782 }
```

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

```
783 \def\zref@ifrefcontainsprop#1#2{%
784   \zref@ifrefundefined{#1}{%
785     \ltx@secondoftwo
786   }{%
787     \expandafter\ZREF@ifrefcontainsprop
788     \csname Z@E@#2\expandafter\endcsname
789     \csname#2\expandafter\expandafter\expandafter\endcsname
790     \expandafter\expandafter\expandafter{%
791       \csname Z@R@#1\endcsname
792     }%
793   }%
794 }
795 \def\ZREF@ifrefcontainsprop#1#2#3{%
796   \expandafter\ifx\expandafter\ZREF@novalue
797   #1#3#2\ZREF@novalue\ZREF@nil\ltx@empty
798   \expandafter\ltx@secondoftwo
799   \else
800   \expandafter\ltx@firstoftwo
801   \fi
802 }
803 \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.

```
804 \def\ZREF@extract#1#2{%
805   \romannumeral0%
806   \ltx@ifundefined{Z@D@#2}{%
807     \expandafter\ltx@space\zref@default
808   }{%
809     \expandafter\expandafter\expandafter\ZREF@@extract
810     \expandafter\expandafter\expandafter{%
811       \csname Z@D@#2\endcsname
812     }{#1}{#2}%
813   }%
814 }
```

`\ZREF@extract`

```
815 \def\ZREF@extract#1#2#3{%
816   \expandafter\expandafter\expandafter\ltx@space
817   \zref@extractdefault{#2}{#3}{#1}%
818 }
```

`\ZREF@wu@extract`

```
819 \def\ZREF@wu@extract#1#2{%
820   \etex@unexpanded\expandafter\expandafter\expandafter{%
821     \ZREF@extract{#1}{#2}%
822   }%
823 }
```

`\zref@extract`

```
824 \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.

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

`\ZREF@wu@extractdefault`

```
839 \def\ZREF@wu@extractdefault#1#2#3{%
840   \etex@unexpanded\expandafter\expandafter\expandafter{%
841     \ZREF@extractdefault{#1}{#2}{#3}%
842   }%
843 }
```

`\zref@extractdefault`

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

`\zref@def@extract`

```
845 \ZREF@Robust\def\zref@def@extract#1{%
846   \zref@wrapper@babel{\ZREF@def@extract{#1}}%
847 }
```

`\ZREF@def@extract`

```
848 \def\ZREF@def@extract#1#2#3{%
849   \zref@refused{#2}%
850   \expandafter\expandafter\expandafter\def
851   \expandafter\expandafter\expandafter#1%
852   \expandafter\expandafter\expandafter{%
853     \zref@extract{#2}{#3}%
854   }%
855 }
```

`\zref@def@extractdefault`

```
856 \ZREF@Robust\def\zref@def@extractdefault#1{%
857   \zref@wrapper@babel{\ZREF@def@extractdefault{#1}}%
858 }
```

`\ZREF@def@extractdefault`

```
859 \def\ZREF@def@extractdefault#1#2#3#4{%
860   \zref@refused{#2}%
861   \expandafter\expandafter\expandafter\def
862   \expandafter\expandafter\expandafter#1%
863   \expandafter\expandafter\expandafter{%
864     \zref@extractdefault{#2}{#3}{#4}%
865   }%
866 }
```

`\ZREF@wrapper@unexpanded`

```
867 \ZREF@Robust{\long\def}\ZREF@wrapper@unexpanded#1{%
868   \let\zref@wrapper@unexpanded\ltx@firstofone
869   \let\zref@getcurrent\ZREF@wu@getcurrent
870   \let\zref@extractdefault\ZREF@wu@extractdefault
871   \let\zref@extract\ZREF@wu@extract
872   #1%
873   \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded
874   \let\zref@getcurrent\ZREF@getcurrent
875   \let\zref@extractdefault\ZREF@extractdefault
876   \let\zref@extract\ZREF@extract
877 }
```

`\zref@wrapper@unexpanded`

```
878 \ltx@ifundefined{etex@unexpanded}{%
879   \let\zref@wrapper@unexpanded\ltx@firstofone
880 }{%
881   \let\zref@wrapper@unexpanded\ZREF@wrapper@unexpanded
882 }
```

6.2.10 Compatibility with babel

`\zref@wrapper@babel`

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

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>

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

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

```
912   \renewcommand*{\thezref@unique}{%
913     zref@number\c@zref@unique
914   }%
```

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>

```
915   \IfFormatAtLeastTF{2020/10/01}
916   {%
917     \AddToHook
918     {include/after}
919     {\setcounter{zref@unique}
920      {\csname fp_to_int:n\endcsname{ceil(\c@zref@unique +100,-2)}}}
921   }{%
922 }{%
923 }
```

6.2.12 Utilities

`\ZREF@number`

```
924 \ltx@ifundefined{numexpr}{%
925   \def\ZREF@number#1{\number#1}%
926 }{%
927   \def\ZREF@number#1{\the\numexpr(#1)\relax}%
928 }
```

6.2.13 Setup

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

```
929 \ZREF@Robust\def\zref@setdefault#1{%
930   \def\zref@default{#1}%
931 }
```

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

```
932 \zref@setdefault{%
933   \nfss@text{\reset@font\bfseries ??}%
934 }
```

Main property list.

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

```
935 \ZREF@Robust\def\zref@setmainlist#1{%
936   \def\ZREF@mainlist{#1}%
937 }
938 \zref@setmainlist{main}
```

Now we create the list.

```
939 \zref@newlist\ZREF@mainlist
```

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

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

`page` the appearance of the page counter

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

Properties

`\ZREF@NewPropAnchor`

```
943 \def\ZREF@NewPropAnchor{%
944   \zref@newprop{anchor}{%
945     \ltx@ifundefined{@currentHref}{-}{\@currentHref}}%
946   }%
947   \global\let\ZREF@NewPropAnchor\relax
948 }
```

`\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`

```
949 \def\ZREF@NewPropTitle{%
950   \gdef\zref@titleref@current{}%
951   \zref@newprop{title}{\zref@titleref@current}}%
952   \global\let\ZREF@NewPropTitle\relax
953 }
```

`\ZREF@NewPropTheotype`

```
954 \def\ZREF@NewPropTheotype{%
955   \zref@newprop{theotype}{}}%
956   \global\let\ZREF@NewPropTheotype\relax
957 }
```

`\ZREF@NewPropPageValue`

```
958 \def\ZREF@NewPropPageValue{%
959   \zref@newprop*{pagevalue}[0]{\number\c@page}}%
960   \global\let\ZREF@NewPropPageValue\relax
961 }
```

Mark successful loading

```
962 \let\ZREF@base@ok=Y
963 </base>
```

6.3 Module user

```
964 <*user>
965 \NeedsTeXFormat{LaTeX2e}
966 \ProvidesPackage{zref-user}%
967   [2026-03-07 v2.37 Module user for zref (H0)]%
968 \RequirePackage{zref-base}[2019/11/29]
969 \ifx\ZREF@base@ok Y%
970 \else
971   \expandafter\endinput
972 \fi
```

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

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

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

```

973 \IfFormatAtLeastTF{2025-06-01}
974 {
975   \newcommand*\zlabel{%
976     \ifx\label@gobble@om
977       \expandafter\ltx@gobble
978     \else
979       \expandafter\zref@wrapper@babel\expandafter\zref@label
980     \fi
981   }
982 }
983 {
984   \newcommand*\zlabel{%
985     \ifx\label\ltx@gobble
986       \expandafter\ltx@gobble
987     \else
988       \expandafter\zref@wrapper@babel\expandafter\zref@label
989     \fi
990   }
991 }%
```

`\zkvlabel`

```

992 \newcommand*\zkvlabel}[1]{%
993   \ifx\label\ltx@gobble
994     \expandafter\ltx@gobblethree
995   \fi
996   \zref@wrapper@babel{\zref@labelbykv{#1}}%
997 }%
```

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

```

998 \newcommand*\zref}[2][default]{% robust because of optional argument
999   \zref@propexists{#1}{%
1000     \zref@wrapper@babel\ZREF@zref{#2}{#1}}%
1001   }%
1002 }%
1003 \def\ZREF@zref#1{%
1004   \zref@refused{#1}%
1005   \zref@extract{#1}%
1006 }%
```

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

```

1007 \ZREF@IfDefinable\zpageref\def{%
1008   {\zref[page]}%
1009 }
```

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

```

1010 \ZREF@IfDefinable\zrefused\def{%
1011   {\zref@refused}%
1012 }

1013 </user>
```

6.4 Module `abspage`

```
1014 ⟨*abspage⟩
1015 \NeedsTeXFormat{LaTeX2e}
1016 \ProvidesPackage{zref-abspage}%
1017 [2026-03-07 v2.37 Module abspage for zref (HO)]%
1018 \RequirePackage{zref-base}[2019/11/29]
1019 \ifx\ZREF@base@ok Y%
1020 \else
1021 \expandafter\endinput
1022 \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.

```
1023 \begingroup
1024 \@ifundefined{c@abspage}
1025 {%
1026 \let\@addtoreset\ltx@gobbletwo
1027 \newcounter{abspage}%
1028 }{}
1029 \endgroup
1030 \setcounter{abspage}{0}%
1031 \AddToHook{shipout/before}{%
1032 \stepcounter{abspage}%
1033 }%
1034 \zref@newprop*{abspage}[0]{\the\ReadonlyShipoutCounter}%
1035 \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.

```
1036 ⟨/abspage⟩
```

6.5 Module `counter`

```
1037 ⟨*counter⟩
1038 \NeedsTeXFormat{LaTeX2e}
1039 \ProvidesPackage{zref-counter}%
1040 [2026-03-07 v2.37 Module counter for zref (HO)]%
1041 \RequirePackage{zref-base}[2019/11/29]
1042 \ifx\ZREF@base@ok Y%
1043 \else
1044 \expandafter\endinput
1045 \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 L^AT_EX 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.

```
1046 \@ifl@t@r\fmtversion{2020-10-01}
1047 {
1048 \zref@newprop{counter}{\@currentcounter}
1049 \zref@addprop\ZREF@mainlist{counter}
1050 }
1051 {
1052 \zref@newprop{counter}{}
1053 \zref@addprop\ZREF@mainlist{counter}
1054 \AtBeginDocument{%
1055 \ZREF@patch{refstepcounter}{%
```

```

1056     \def\refstepcounter#1{%
1057         \zref@setcurrent{counter}{#1}%
1058         \ZREF@org@refstepcounter{#1}%
1059     }%
1060 }%
1061 }
1062 }
1063 </counter>

```

6.6 Module lastpage

```

1064 <*lastpage>
1065 \NeedsTeXFormat{LaTeX2e}
1066 \ProvidesPackage{zref-lastpage}%
1067 [2026-03-07 v2.37 Module lastpage for zref (HO)]%
1068 \RequirePackage{zref-base}[2019/11/29]
1069 \RequirePackage{zref-abspage}[2019/11/29]
1070 \ifx\ZREF@base@ok Y%
1071 \else
1072 \expandafter\endinput
1073 \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.

```

1074 \zref@newlist{LastPage}
1075 \@ifl@t@r\fmtversion{2025-06-01}{%
1076 \AddToHook {enddocument/afterlastpage}{%
1077 \if@filesw
1078 \begingroup
1079 \toks@\expandafter\expandafter\expandafter{%
1080 \expandafter\Z@L@main
1081 \Z@L@LastPage
1082 }%
1083 \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
1084 \endgroup
1085 \fi
1086 }}
1087 {\AddToHook {enddocument/afterlastpage}{%
1088 \if@filesw
1089 \begingroup
1090 \advance\c@page\m@ne
1091 \toks@\expandafter\expandafter\expandafter{%
1092 \expandafter\Z@L@main
1093 \Z@L@LastPage
1094 }%
1095 \expandafter\zref@wrapper@immediate\expandafter{%
1096 \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
1097 }%
1098 \endgroup
1099 \fi
1100 }}

```

\zref@iflastpage

```

1101 \def\zref@iflastpage#1{%
1102 \ifnum\zref@extractdefault{#1}{abspage}{-1}=%
1103 \zref@extractdefault{LastPage}{abspage}{-2} %
1104 \expandafter\ltx@firstoftwo
1105 \else
1106 \expandafter\ltx@secondoftwo
1107 \fi
1108 }

```

`\ziflastpage`

```
1109 \ZREF@ifDefinable\ziflastpage\def{%  
1110   {\zref@wrapper@babel\ZREF@iflastpage}%  
1111 }
```

`ZREF@iflastpage`

```
1112 \def\ZREF@iflastpage#1{%  
1113   \zref@refused{LastPage}%  
1114   \zref@refused{#1}%  
1115   \zref@iflastpage{#1}%  
1116 }
```

```
1117 </lastpage>
```

6.7 Module `thepage`

```
1118 <*thepage>  
1119 \NeedsTeXFormat{LaTeX2e}  
1120 \ProvidesPackage{zref-thepage}%  
1121 [2026-03-07 v2.37 Module thepage for zref (HO)]%  
1122 \RequirePackage{zref-base}[2019/11/29]  
1123 \ifx\ZREF@base@ok Y%  
1124 \else  
1125   \expandafter\endinput  
1126 \fi  
  
1127 \RequirePackage{atbegshi}[2011/10/05]  
1128 \RequirePackage{zref-abspage}[2019/11/29]  
  
1129 \zref@newlist{thepage}  
1130 \zref@addprop{thepage}{page}  
1131 \ZREF@NewPropPageValue
```

`\zref@thepage@atbegshi@hook`

```
1132 \let\zref@thepage@atbegshi@hook\ltx@empty  
  
1133 \zref@addprop{thepage}{pagevalue}  
1134 \AddToHook{shipout/before}{%  
1135   \AtBeginShipoutAddToBox{%  
1136     \zref@thepage@atbegshi@hook  
1137     \zref@labelbylist{thepage\the\value{abspage}}{thepage}%  
1138   }%  
1139 }
```

`\zref@thepage@name`

```
1140 \ltx@ifUndefined{numexpr}{%  
1141   \def\zref@thepage@name#1{thepage\number#1}%  
1142 }{%  
1143   \def\zref@thepage@name#1{thepage\the\numexpr#1}%  
1144 }
```

`\zref@thepage`

```
1145 \def\zref@thepage#1{%  
1146   \zref@extract{\zref@thepage@name{#1}}{page}%  
1147 }%
```

`\zref@thepage@refused`

```
1148 \ZREF@Robust\def\zref@thepage@refused#1{%  
1149   \zref@refused{\zref@thepage@name{#1}}%  
1150 }%
```

\zthepage

```
1151 \ZREF@ifDefinable\zthepage\def{%
1152   #1{%
1153     \zref@thepage@refused{#1}%
1154     \zref@thepage{#1}%
1155   }%
1156 }

1157 </thepage>
```

6.8 Module nextpage

```
1158 <*nextpage>
1159 \NeedsTeXFormat{LaTeX2e}
1160 \ProvidesPackage{zref-nextpage}%
1161 [2026-03-07 v2.37 Module nextpage for zref (HO)]%
1162 \RequirePackage{zref-base}[2019/11/29]
1163 \ifx\ZREF@base@ok Y%
1164 \else
1165   \expandafter\endinput
1166 \fi

1167 \RequirePackage{zref-abspage}[2019/11/29]
1168 \RequirePackage{zref-thepage}[2019/11/29]
1169 \RequirePackage{zref-lastpage}[2019/11/29]
1170 \RequirePackage{uniquecounter}[2009/12/18]

1171 \UniqueCounterNew{znextpage}
1172
1173 \newcommand*{\znextpagesetup}{%
1174   \afterassignment\ZREF@np@setup@i
1175   \def\ZREF@np@call@unknown##1%
1176 }
1177 \def\ZREF@np@setup@i{%
1178   \afterassignment\ZREF@np@setup@ii
1179   \def\ZREF@np@call@nonext##1%
1180 }
1181 \def\ZREF@np@setup@ii{%
1182   \def\ZREF@np@call@next##1%
1183 }
1184 \def\ZREF@np@call@unknown#1{#1}
1185 \def\ZREF@np@call@nonext#1{#1}
1186 \def\ZREF@np@call@next#1{#1}
1187 \ZREF@ifDefinable\znextpage\def{%
1188   {\UniqueCounterCall{znextpage}{\ZREF@nextpage}}%
1189 }%
1190 \newcommand*{\znonextpagename}{}
1191 \newcommand*{\zunknownnextpagename}{\Z@D@page}
1192 \def\ZREF@nextpage#1{%
1193   \begingroup
1194     \def\ZREF@refname@this{zref@np#1}%
1195     \zref@labelbyprops\ZREF@refname@this{abspage}%
1196     \chardef\ZREF@call=0 % unknown
1197     \ZREF@ifrefundefined\ZREF@refname@this{%
1198       }{%
1199         \edef\ZREF@pagenum@this{%
1200           \zref@extractdefault\ZREF@refname@this{abspage}{0}%
1201         }%
1202         \edef\ZREF@refname@next{%
1203           \zref@thepage@name{%
1204             \the\numexpr\ZREF@pagenum@this+1%
1205           }%
1206         }%
1207         \ifnum\ZREF@pagenum@this>0 %
```

```

1208     \ZREF@ifrefundefined{LastPage}{%
1209     \zref@ifrefundefined\ZREF@refname@next{%
1210     }{%
1211     \chardef\ZREF@call=2 % next page
1212     }%
1213 }{%
1214     \edef\ZREF@pagenum@last{%
1215     \zref@extractdefault{LastPage}{abspage}{0}%
1216     }%
1217     \ifnum\ZREF@pagenum@this<\ZREF@pagenum@last\ltx@space
1218     \ZREF@ifrefundefined\ZREF@refname@next{%
1219     }{%
1220     \chardef\ZREF@call=2 % next page
1221     }%
1222     \else
1223     \ifnum\ZREF@pagenum@this=\ZREF@pagenum@this\ltx@space
1224     \chardef\ZREF@call=1 % no next page
1225     \fi
1226     \fi
1227     }%
1228     \fi
1229 }%
1230 \edef\x{%
1231     \endgroup
1232     \ifcase\ZREF@call
1233     \noexpand\ZREF@np@call@unknown{%
1234     \noexpand\zunknownnextpagename
1235     }%
1236     \or
1237     \noexpand\ZREF@np@call@nonext{%
1238     \noexpand\znonextpagename
1239     }%
1240     \else
1241     \noexpand\ZREF@np@call@next{%
1242     \noexpand\zref@extract{\ZREF@refname@next}{page}%
1243     }%
1244     \fi
1245     }%
1246     \x
1247 }
1248 </nextpage>

```

6.9 Module `totpages`

```

1249 <*totpages>
1250 \NeedsTeXFormat{LaTeX2e}
1251 \ProvidesPackage{zref-totpages}%
1252 [2026-03-07 v2.37 Module totpages for zref (HO)]%
1253 \RequirePackage{zref-base}[2019/11/29]
1254 \ifx\ZREF@base@ok Y%
1255 \else
1256 \expandafter\endinput
1257 \fi

```

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

```

1258 \RequirePackage{zref-abspage}[2019/11/29]
1259 \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.

```

1260 \newcommand*{\ztotpages}{%
1261     \zref@extractdefault{LastPage}{abspage}{0}%
1262 }

```

Also we mark the reference LastPage as used:

```
1263 \AtBeginDocument{%
1264   \zref@refused{LastPage}%
1265 }
1266 </totpages>
```

6.10 Module pagelayout

```
1267 <*pagelayout>
1268 \NeedsTeXFormat{LaTeX2e}
1269 \ProvidesPackage{zref-pagelayout}%
1270 [2026-03-07 v2.37 Module pagelayout for zref (H0)]%
1271 \RequirePackage{zref-base}[2019/11/29]
1272 \ifx\ZREF@base@ok Y%
1273 \else
1274   \expandafter\endinput
1275 \fi

1276 \RequirePackage{zref-thepage}[2019/11/29]
1277 \RequirePackage{iftex}[2019/11/07]%
```

6.10.1 Define layout properties

```
1278 \def\ZREF@temp#1{%
1279   \begingroup
1280     \escapechar=-1 %
1281     \ltx@ifundefined{\string#1}{\endgroup}{%
1282       \edef\x{%
1283         \endgroup
1284         \noexpand\zref@newprop*{\string#1}%
1285           [\noexpand\number\noexpand#1]% hash-ok
1286           {\noexpand\number\noexpand#1}%
1287         \noexpand\zref@addprop{thepage}{\string#1}%
1288       }%
1289     \x
1290   }%
1291 }

1292 \ZREF@temp\mag
1293 \ZREF@temp\paperwidth
1294 \ZREF@temp\paperheight
1295 \ZREF@temp\stockwidth % memoir.cls, crop.sty
1296 \ZREF@temp\stockheight % memoir.cls, crop.sty
1297 \ZREF@temp\mediawidth % VTeX
1298 \ZREF@temp\mediaheight % VTeX
1299 \ifluatex
1300 \zref@newprop*{pdfvorigin}%
1301   [\number\pdfvariable vorigin]% hash-ok
1302   {\number\pdfvariable vorigin}%
1303 \zref@addprop{thepage}{pdfvorigin}
1304 \zref@newprop*{pdfhorigin}%
1305   [\number\pdfvariable horigin]% hash-ok
1306   {\number\pdfvariable horigin}%
1307 \zref@addprop{thepage}{pdfhorigin}
1308 \zref@newprop*{pdfpageheight}%
1309   [\number\pageheight]% hash-ok
1310   {\number\pageheight}%
1311 \zref@addprop{thepage}{pdfpageheight}
1312 \zref@newprop*{pdfpagewidth}%
1313   [\number\pagewidth]% hash-ok
1314   {\number\pagewidth}%
1315 \zref@addprop{thepage}{pdfpagewidth}
1316 \else
1317 \ZREF@temp\pdfpagewidth
```

```

1318 \ZREF@temp\pdfpageheight
1319 \ZREF@temp\pdfhorigin
1320 \ZREF@temp\pdfvorigin
1321 \fi
1322 \ZREF@temp\hoffset
1323 \ZREF@temp\voffset
1324 \ZREF@temp\topmargin
1325 \ZREF@temp\oddsidemargin
1326 \ZREF@temp\evensidemargin
1327 \ZREF@temp\textwidth
1328 \ZREF@temp\textheight
1329 \ZREF@temp\headheight
1330 \ZREF@temp\headsep
1331 \ZREF@temp\footskip
1332 \ZREF@temp\marginparwidth
1333 \ZREF@temp\marginparsep
1334 \ZREF@temp\columnwidth
1335 \ZREF@temp\columnsep
1336 \ZREF@temp\trimedge % memoir.cls
1337 \ZREF@temp\spinemargin % memoir.cls
1338 \ZREF@temp\foremargin % memoir.cls
1339 \ZREF@temp\trimtop % memoir.cls
1340 \ZREF@temp\uppermargin % memoir.cls
1341 \ZREF@temp\headmargin % memoir.cls

1342 \IfFormatAtLeastTF{2020/10/01}
1343 {
1344   \zref@newprop*{outputboxwd}[0pt]{\ShipoutBoxWidth}
1345   \zref@newprop*{outputboxht}[0pt]{\ShipoutBoxHeight}
1346   \zref@newprop*{outputboxdp}[0pt]{\ShipoutBoxDepth}
1347 }
1348 {
1349   \zref@newprop*{outputboxwd}[0pt]{\AtBeginShipoutBoxWidth}
1350   \zref@newprop*{outputboxht}[0pt]{\AtBeginShipoutBoxHeight}
1351   \zref@newprop*{outputboxdp}[0pt]{\AtBeginShipoutBoxDepth}
1352 }

1353 \zref@addprops{thepage}{outputboxwd,outputboxht,outputboxdp}

\ifZREF@pl@list
1354 \ltx@newif\ifZREF@pl@list

\zref@listpagelayout
1355 \ZREF@IfDefinable\zlistpagelayout\def{%
1356   {\global\ZREF@pl@listtrue}%
1357 }

\ZREF@pl@AfterLastShipout
1358 \def\ZREF@pl@AfterLastShipout{%
1359   \ifZREF@pl@list
1360     \edef\ZREF@page@max{\the\value{abspage}}%
1361     \ltx@ifundefined{ZREF@org@testdef}{%
1362       \let\ZREF@org@testdef\@testdef
1363       \def\@testdef##1##2##3{%
1364         \ZREF@org@testdef{##1}{##2}{##3}%
1365         \def\ZREF@temp{##1}%
1366         \ifx\ZREF@temp\ZREF@RefPrefix
1367           \expandafter\gdef\csname##1@##2\endcsname{##3}%
1368         \fi
1369       }%
1370     }{}%
1371     \AddToHook{enddocument/afteraux}{\ZREF@pl@AtVeryEnd}%
1372   \fi
1373 }

```

\ZREF@pl@AtVeryEnd

```
1374 \def\ZREF@pl@AtVeryEnd{%
1375   \beginingroup
1376     \toks@{Page layout parameters:\MessageBreak}%
1377     \count@=1 %
1378     \ZREF@pl@ListPage
1379     \edef\x{\endgroup
1380       \noexpand\@PackageInfoNoLine{zref-pagelayout}{\the\toks@}%
1381     }%
1382   \x
1383 }
```

\ZREF@pl@ListPage

```
1384 \def\ZREF@pl@ListPage{%
1385   \edef\x{%
1386     \toks@={%
1387       \the\toks@
1388       Page \the\count@:\noexpand\MessageBreak
1389       \zref@ifrefundefined{thepage\the\count@}{-}{%
1390         \ltx@space\ltx@space mag = %
1391         \zref@extract{thepage\the\count@}{mag}%
1392         \noexpand\MessageBreak
1393         \ZREF@pl@ListEntry{paperwidth}%
1394         \ZREF@pl@ListEntry{paperheight}%
1395         \ZREF@pl@ListEntry{stockwidth}%
1396         \ZREF@pl@ListEntry{stockheight}%
1397         \ZREF@pl@ListEntry{mediawidth}%
1398         \ZREF@pl@ListEntry{mediaheight}%
1399         \ZREF@pl@ListEntry{pdfpagewidth}%
1400         \ZREF@pl@ListEntry{pdfpageheight}%
1401         \ZREF@pl@ListEntry{pdfhorigin}%
1402         \ZREF@pl@ListEntry{pdfvorigin}%
1403         \ZREF@pl@ListEntry{hoffset}%
1404         \ZREF@pl@ListEntry{voffset}%
1405         \ZREF@pl@ListEntry{topmargin}%
1406         \ZREF@pl@ListEntry{oddsidemargin}%
1407         \ZREF@pl@ListEntry{evensidemargin}%
1408         \ZREF@pl@ListEntry{textwidth}%
1409         \ZREF@pl@ListEntry{textheight}%
1410         \ZREF@pl@ListEntry{headheight}%
1411         \ZREF@pl@ListEntry{headsep}%
1412         \ZREF@pl@ListEntry{footskip}%
1413         \ZREF@pl@ListEntry{marginparwidth}%
1414         \ZREF@pl@ListEntry{marginparsep}%
1415         \ZREF@pl@ListEntry{columnwidth}%
1416         \ZREF@pl@ListEntry{columnsep}%
1417         \ZREF@pl@ListEntry{trimedge}%
1418         \ZREF@pl@ListEntry{spinemargin}%
1419         \ZREF@pl@ListEntry{foremargin}%
1420         \ZREF@pl@ListEntry{trimtop}%
1421         \ZREF@pl@ListEntry{uppermargin}%
1422         \ZREF@pl@ListEntry{headmargin}%
1423       }%
1424     }%
1425   }\x
1426   \ifnum\ZREF@page@max>\count@
1427     \advance\count@ by\ltx@one
1428   \else
1429     \expandafter\ltx@gobble
1430   \fi
1431   \ZREF@pl@ListPage
1432 }
```

\ZREF@pl@ListEntry

```
1433 \def\ZREF@pl@ListEntry#1{%
1434   \zref@ifpropundefined{#1}{%
1435   }{%
1436     \zref@ifrefcontainsprop{thepage\the\count@}{#1}{%
1437     \ltx@space\ltx@space#1 = %
1438     \zref@extract{thepage\the\count@}{#1}sp = %
1439     \the\dimexpr\zref@extract{thepage\the\count@}{#1}sp\relax
1440     \noexpand\MessageBreak
1441     }{%}
1442   }%
1443 }

1444 \AddToHook {enddocument/afterlastpage}{%
1445   \ZREF@pl@AfterLastShipout
1446 }

1447 </pagelayout>
```

6.11 Module pageattr

```
1448 <*pageattr>
1449 \NeedsTeXFormat{LaTeX2e}
1450 \ProvidesPackage{zref-pageattr}%
1451   [2026-03-07 v2.37 Module pageattr for zref (HO)]%
1452 \RequirePackage{zref-base}[2019/11/29]
1453 \ifx\ZREF@base@ok Y%
1454 \else
1455   \expandafter\endinput
1456 \fi

1457 \RequirePackage{iftex}[2019/11/07]%

1458 \let\ZREF@temp=N%
1459 \ifluatex
1460 \expandafter\@firstoftwo
1461 \else
1462 \expandafter\@secondoftwo
1463 \fi
1464 {%\luatex
1465 \RequirePackage{zref-thepage}[2019/11/29]
1466 \RequirePackage{zref-lastpage}[2019/11/29]%
1467 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfvariable pageattr}}%
1468 \zref@addprop{thepage}{pdfpageattr}%
1469 \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfvariable pagesattr}}%
1470 \zref@addprop{LastPage}{pdfpagesattr}%
1471 \let\ZREF@temp=Y%
1472 }
1473 {%other
1474 \ltx@ifundefined{pdfpageattr}{%
1475   \@PackageInfoNoLine{zref-pageattr}{%
1476     \string\pdfpageattr\space is not available%
1477   }%
1478   \def\zref@pdfpageattr#1{}%
1479   \def\zref@pdfpageattr@used#1{}%
1480 }{%
1481 \RequirePackage{zref-thepage}[2019/11/29]%
1482 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfpageattr}}%
1483 \zref@addprop{thepage}{pdfpageattr}%
1484 \let\ZREF@temp=Y%
1485 }
1486 \ltx@ifundefined{pdfpagesattr}{%
1487   \@PackageInfoNoLine{zref-pageattr}{%
1488     \string\pdfpagesattr\space is not available%
```

```

1489 }%
1490 \def\zref@pdfpagesattr{}%
1491 \def\zref@pdfpagesattr@used{}%
1492 }{%
1493 \RequirePackage{zref-lastpage}[2019/11/29]%
1494 \zref@newprop*{pdfpagesattr}[]{\zref@hex{the\pdfpagesattr}}%
1495 \zref@addprop{LastPage}{pdfpagesattr}%
1496 \let\ZREF@temp=Y%
1497 }%
1498 }%

1499 \ifx\ZREF@temp N%
1500 \expandafter\endinput
1501 \fi

1502 \RequirePackage{zref-abspage}[2019/11/29]
1503 \RequirePackage{pdftexcmds}[2010/04/01]
1504 \let\ZREF@temp=Y%
1505 \ltx@ifundefined{pdf@escapehex}{\let\ZREF@temp=N}{%
1506 \ltx@ifundefined{pdf@unescapehex}{\let\ZREF@temp=N}{%
1507 \ifx\ZREF@temp N%
1508 \let\zref@hex\ltx@firstofone
1509 \let\zref@unhex\ltx@firstofone
1510 \else
1511 \let\zref@hex\pdf@escapehex
1512 \let\zref@unhex\pdf@unescapehex
1513 \fi

\ifZREF@pa@list
1514 \ltx@newif\ifZREF@pa@list

\zref@listpageattr
1515 \ZREF@ifDefinable\zlistpageattr\def{%
1516 {\ZREF@pa@listtrue}%
1517 }

\ZREF@pa@AfterLastShipout
1518 \def\ZREF@pa@AfterLastShipout{%
1519 \ifZREF@pa@list
1520 \edef\ZREF@page@max{\the\value{abspage}}%
1521 \ltx@ifundefined{ZREF@org@testdef}{%
1522 \let\ZREF@org@testdef\@testdef
1523 \def\@testdef##1##2##3{%
1524 \ZREF@org@testdef{##1}{##2}{##3}%
1525 \def\ZREF@temp{##1}%
1526 \ifx\ZREF@temp\ZREF@RefPrefix
1527 \expandafter\xdef\csname##1@##2\endcsname{##3}%
1528 \fi
1529 }%
1530 }{}%
1531 \AddToHook{enddocument/afteraux}{\ZREF@pa@AtVeryEnd}%
1532 \fi
1533 }

\ZREF@pa@AtVeryEnd
1534 \let\ZREF@temp=Y%
1535 \ltx@ifundefined{pdfpageattr}{}{\let\ZREF@temp=N}
1536 \ifluatex \let\ZREF@temp=N \fi
1537 \ifx\ZREF@temp Y
1538 \expandafter\@firstoftwo
1539 \else
1540 \expandafter\@secondoftwo
1541 \fi
1542 {%

```

```

1543 \def\ZREF@pa@AtVeryEnd{}%
1544 }
1545 {%
1546 \def\ZREF@pa@AtVeryEnd{%
1547 \begingroup
1548 \toks@{List of \ltx@backslashchar
1549 \ifluatex pdfvariable\else pdf\fi
1550 pdfpageattr:\MessageBreak}%
1551 \count@=1 %
1552 \ZREF@pa@ListPage
1553 \edef\x{\endgroup
1554 \noexpand\@PackageInfoNoLine{zref-pageattr}{}%
1555 \the\toks@
1556 }%
1557 }%
1558 \x
1559 }%

```

\zref@pageattr

```

1560 \def\zref@pdfpageattr#1{%
1561 \zref@unhex{%
1562 \zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}%
1563 }%
1564 }
1565 % compatibility, \zref@pageattr was defined in older versions
1566 \let\zref@pageattr\zref@pdfpageattr

```

\zref@pageattr@used

```

1567 \ZREF@Robust\def\zref@pageattr@used#1{%
1568 \zref@refused{thepage\ZREF@number{#1}}}%
1569 }

```

\ZREF@pa@ListPage

```

1570 \def\ZREF@pa@ListPage{%
1571 \edef\x{%
1572 \toks@={%
1573 \the\toks@
1574 Page \the\count@:%
1575 \noexpand\MessageBreak
1576 \zref@ifrefundefined{thepage\the\count@}{}%
1577 <<\zref@pdfpageattr\count@>>%
1578 \noexpand\MessageBreak
1579 }%
1580 }%
1581 }\x
1582 \ifnum\ZREF@page@max>\count@
1583 \advance\count@ by\ltx@one
1584 \else
1585 \expandafter\ltx@gobble
1586 \fi
1587 \ZREF@pa@ListPage
1588 }%
1589 }

```

```

1590 \let\ZREF@temp=Y%
1591 \ltx@ifundefined{pdfpagesattr}{\let\ZREF@temp=N}
1592 \ifluatex \let\ZREF@temp=N \fi
1593 \ifx\ZREF@temp N
1594 \expandafter\@firstofone
1595 \fi
1596 {%

```

`\zref@pdfpagesattr`

```
1597 \def\zref@pdfpagesattr{%
1598   \zref@unhex{%
1599     \zref@extract{LastPage}{pdfpagesattr}%
1600   }%
1601 }%
```

`\zref@pdfpagesattr@used`

```
1602 \ZREF@Robust\def\zref@pdfpagesattr@used{%
1603   \zref@refused{LastPage}%
1604 }%
1605 \ltx@LocalAppendToMacro\ZREF@pa@AtVeryEnd{%
1606   \@PackageInfoNoLine{zref-pageattr}{%
1607     \ltx@backslashchar
1608     \ifluatex pdfvariable\else pdf\fi
1609     pagesattr:\MessageBreak
1610     <<\zref@pdfpagesattr>>%
1611     \MessageBreak
1612   }%
1613 }%
1614 }
1615 \AddToHook {enddocument/afterlastpage}{%
1616   \ZREF@pa@AfterLastShipout
1617 }
1618 </pageattr>
```

6.12 Module marks

```
1619 <*marks>
1620 \NeedsTeXFormat{LaTeX2e}
1621 \ProvidesPackage{zref-marks}%
1622 [2026-03-07 v2.37 Module marks for zref (HO)]%
1623 \RequirePackage{zref-base}[2019/11/29]
1624 \ifx\ZREF@base@ok Y%
1625 \else
1626   \expandafter\endinput
1627 \fi
1628 \newcommand*{\zref@marks@register}[3][ ]{%
1629   \edef\ZREF@TempName{#1}%
1630   \edef\ZREF@TempNum{\ZREF@number{#2}}%
1631   \ifnum\ZREF@TempNum<\ltx@zero %
1632     \PackageError\ZREF@name{%
1633       \string\zref@marks@register\ltx@space is called with invalid%
1634       \MessageBreak
1635       marks register number (\ZREF@TempNum)%
1636     }{%
1637       Use ‘0’ or the command, defined by \string\newmarks.\MessageBreak
1638     }%
1639   }%
1640 \else
1641   \ifx\ZREF@TempName\ltx@empty
1642     \edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}%
1643   \else
1644     \edef\ZREF@TempName{marks\ZREF@TempName}%
1645   \fi
1646   \ZREF@MARKS@DefineProp{top}%
1647   \ZREF@MARKS@DefineProp{first}%
1648   \ZREF@MARKS@DefineProp{bot}%
1649   \kv@parse{#3}{%
1650     \ifx\kv@value\relax
1651       \def\kv@value{top,first,bot}%

```

```

1652     \fi
1653     \edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}%
1654     \ifx\ZREF@temp\ZREF@STAR
1655         \edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}%
1656         \zref@newlist\kv@key
1657     \fi
1658     \expandafter\comma@parse\expandafter{\kv@value}{%
1659         \ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else
1660             \ifx\comma@entry\ZREF@NAME@first 1\else
1661                 \ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space
1662         \PackageWarning{zref-marks}{%
1663             Use 'top', 'first' or 'bot' for the list values%
1664             \MessageBreak
1665             in the third argument of \string\zref@marks@register.%
1666             \MessageBreak
1667             Ignoring unknown value '\comma@entry'%
1668         }%
1669     \else
1670         \zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}%
1671     \fi
1672     \ltx@gobble
1673 }%
1674 \ltx@gobbletwo
1675 }%
1676 \fi
1677 }
1678 \def\ZREF@STAR{*}
1679 \def\ZREF@NAME@top{top}
1680 \def\ZREF@NAME@first{first}
1681 \def\ZREF@NAME@bot{bot}
1682 \def\ZREF@MARKS@DefineProp#1{%
1683     \zref@ifpropundefined{#1\ZREF@TempName}{%
1684         \ifnum\ZREF@TempNum=\ltx@zero
1685             \begingroup
1686                 \edef\x{\endgroup
1687                     \noexpand\zref@newprop*{#1\ZREF@TempName} [] {%
1688                         \expandafter\noexpand\csname#1mark\endcsname
1689                     }%
1690                 }%
1691             \x
1692         \else
1693             \begingroup
1694                 \edef\x{\endgroup
1695                     \noexpand\zref@newprop*{#1\ZREF@TempName} [] {%
1696                         \expandafter\noexpand\csname#1marks\endcsname
1697                         \ZREF@TempNum
1698                     }%
1699                 }%
1700             \x
1701         \fi
1702     }{%
1703         \PackageWarning{zref-marks}{%
1704             \string\zref@marks@register\ltx@space does not generate the%
1705             \MessageBreak
1706             new property '#1\ZREF@TempName', because\MessageBreak
1707             it is already defined%
1708         }%
1709     }%
1710 }
1711 </marks>

```

6.13 Module runs

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

```
1712 <*runs>
1713 \NeedsTeXFormat{LaTeX2e}
1714 \ProvidesPackage{zref-runs}%
1715 [2026-03-07 v2.37 Module runs for zref (H0)]%

\zruns
1716 \providecommand*{\zruns}{0}%
1717 \AtBeginDocument{%
1718   \edef\zruns{\number\numexpr\zruns+1}%
1719   \begingroup
1720     \def\on@line{}%
1721     \PackageInfo{zref-runs}{LaTeX runs: \zruns}%
1722     \if@filesw
1723       \immediate\write\@mainaux{%
1724         \string\gdef\string\zruns{\zruns}%
1725       }%
1726     \fi
1727   \endgroup
1728 }

1729 </runs>
```

6.14 Module perpage

```
1730 <*perpage>
1731 \NeedsTeXFormat{LaTeX2e}
1732 \ProvidesPackage{zref-perpage}%
1733 [2026-03-07 v2.37 Module perpage for zref (H0)]%
1734 \RequirePackage{zref-base}[2019/11/29]
1735 \ifx\ZREF@base@ok Y%
1736 \else
1737   \expandafter\endinput
1738 \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.

```
1739 \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.

```
1740 \ZREF@NewPropPageValue
1741 \zref@newlist{perpage}
1742 \zref@addprops{perpage}{abspage,page,pagevalue}
```

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

```
1743 \newcounter{zpage}
```

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

```
1744 \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`.

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

```

1750 \ZREF@org@stepcounter{#1}%
1751 }

If amstext is loaded it overwrites the definition (or we overwrite their definition) so we account for this by using a package hook, see https://github.com/ho-tex/zref/issues/11.
1752 \IfFormatAtLeastTF{2020/10/01}
1753 {
1754   \AddToHook{package/amstext/after}
1755   {\def\stepcounter#1{%
1756     \iffirstchoice@
1757     \ifcsname @stepcounterhook@#1\endcsname
1758     \csname @stepcounterhook@#1\endcsname
1759     \fi
1760     \addtocounter{#1}\@ne
1761     \begingroup \let\@elt\@stpelt \csname cl@#1\endcsname \endgroup
1762     \fi
1763   }}
1764 }{}
1765 %   \end{macrocode}
1766 %
1767 %
1768 %   \cs{@stpelt} must be adapted due to the change in latex
1769 %   2015-01, see https://github.com/ho-tex/zref/issues/26
1770 %   \begin{macrocode}
1771 \let\ZREF@org@@stpelt\@stpelt
1772 \def\@stpelt#1{%
1773   \ifcsname ZREF@perpage@#1\endcsname
1774   \begingroup
1775     \let\stepcounter\ZREF@org@stepcounter
1776     \ZREF@org@@stpelt{#1}%
1777   \endgroup
1778   \expandafter\ltx@gobbletwo
1779   \fi
1780   \ZREF@org@@stpelt{#1}%
1781 }

```

`\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.

```

1782 \ZREF@IfDefinable\zmakeperpage\def{%
1783   {%
1784     \@ifnextchar[\ZREF@makeperpage@opt{\ZREF@@makeperpage[\ltx@zero]}%
1785   }%
1786 }

```

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

```

1787 \def\ZREF@makeperpage@opt[#1]{%
1788   \begingroup
1789     \edef\x{\endgroup
1790       \noexpand\ZREF@@makeperpage[\number\numexpr#1-1\relax]}%
1791   }%
1792   \x
1793 }

1794 \def\ZREF@@makeperpage[#1]#2{%
1795   \@ifundefined{@stepcounterhook@#2}{%
1796     \expandafter\gdef\csname @stepcounterhook@#2\endcsname{}%
1797   }{}%
1798   \expandafter\gdef\csname ZREF@perpage@#2\endcsname{%
1799     \ZREF@@perpage@step{#2}{#1}%
1800   }%

```

```

1801 \expandafter\g@addto@macro\csname @stepcounterhook@#2\endcsname{%
1802   \ifcsname ZREF@perpage@#2\endcsname
1803     \csname ZREF@perpage@#2\endcsname
1804   \fi
1805 }%
1806 }

```

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

```

1807 \def\ZREF@@perpage@step#1#2{%
First the reference is generated.
1808 \global\advance\c@zref@unique\ltx@one
1809 \begingroup
1810 \expandafter
1811 \zref@labelbylist\expandafter{\thezref@unique}{perpage}%

```

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

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

```

1812 \zref@ifrefundefined\thezref@unique{%
1813   \global\c@zpage=\c@page
1814   \global\let\thezpage\thepage
1815   \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1816     \number\c@abspage
1817   }%
1818 }{%

```

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

```

1819   \global\c@zpage=\zref@extract\thezref@unique{pagevalue}\relax
1820   \xdef\thezpage{\noexpand\zref@extract{\thezref@unique}{page}}%
1821   \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1822     \zref@extractdefault\thezref@unique
1823       {abspage}{\number\c@abspage}%
1824   }%
1825 }%

```

Page changes are detected by a changed absolute page number.

```

1826 \expandafter\ifx\csname ZREF@abspage@#1\expandafter\endcsname
1827   \csname ZREF@currentabspage@#1\endcsname
1828 \else
1829   \global\csname c@#1\endcsname=#2\relax
1830   \global\expandafter\let
1831     \csname ZREF@currentabspage@#1\expandafter\endcsname
1832     \csname ZREF@abspage@#1\endcsname
1833 \fi
1834 \endgroup
1835 }

```

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

```

1836 \ZREF@IfDefinable\zunmakeperpage\def{%
1837   #1{%
1838     \global\expandafter
1839     \let\csname ZREF@perpage@#1\endcsname\@undefined
1840   }%
1841 }

1842 </perpage>

```

6.15 Module `titleref`

```

1843 <*titleref>
1844 \NeedsTeXFormat{LaTeX2e}
1845 \ProvidesPackage{zref-titleref}%

```

```

1846 [2026-03-07 v2.37 Module titleref for zref (H0)]%
1847 \RequirePackage{zref-base}[2019/11/29]
1848 \ifx\ZREF@base@ok Y%
1849 \else
1850 \expandafter\endinput
1851 \fi
1852 \RequirePackage{getttitlestring}[2009/12/08]

```

6.15.1 Implementation

```
1853 \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.

```

1854 \ZREF@NewPropTitle
1855 \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.

```
1856 \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`.

```
1857 \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.

```

1858 \ZREF@Robust\def\zref@titleref@cleanup#1{%
1859   \begingroup
1860   \toks@\expandafter{%
1861     \ZREF@titleref@hook
1862     #1%
1863   }%
1864   \expandafter\endgroup
1865   \expandafter\def\expandafter\ZREF@titleref@hook\expandafter{%
1866     \the\toks@
1867   }%
1868 }%

```

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

```

1869 \newif\ifzref@titleref@stripperperiod
1870 \zref@titleref@stripperperiodtrue

```

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

```

1871 \ZREF@Robust\def\zref@titleref@setcurrent#1{%
1872   \ifzref@titleref@expand
1873     \GetTitleStringExpand{#1}%

```

```

1874 \else
1875   \GetTitleStringNonExpand{#1}%
1876 \fi
1877 \edef\zref@titleref@current{%
1878   \detokenize\expandafter{\GetTitleStringResult}%
1879 }%
1880 \ifzref@titleref@stripperperiod
1881   \edef\zref@titleref@current{%
1882     \expandafter\ZREF@stripperperiod\zref@titleref@current
1883     \ltx@empty.\ltx@empty\@nil
1884   }%
1885 \fi
1886 }%
1887 \GetTitleStringDisableCommands{%
1888   \ZREF@titleref@hook
1889 }

```

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

```

1890 \def\ZREF@stripperperiod#1.\ltx@empty#2\@nil{#1}%

```

6.15.2 User interface

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

```

1891 \define@key{ZREF@TR}{expand}[true]{%
1892   \csname zref@titleref@expand#1\endcsname
1893 }%
1894 \define@key{ZREF@TR}{stripperperiod}[true]{%
1895   \csname zref@titleref@stripperperiod#1\endcsname
1896 }%
1897 \define@key{ZREF@TR}{cleanup}{%
1898   \zref@titleref@cleanup{#1}%
1899 }%
1900 \define@key{ZREF@TR}{title}{%
1901   \def\zref@titleref@current{#1}%
1902 }%
1903 \ZREF@IfDefinable\ztitlerefsetup\def{%
1904   {\kvsetkeys{ZREF@TR}}%
1905 }%

```

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

```

1906 \ZREF@IfDefinable\ztitleref\def{%
1907   {\zref@wrapper@babel\ZREF@titleref}%
1908 }%
1909 \def\ZREF@titleref#1{%
1910   \begingroup
1911     \zref@refused{#1}%
1912     \let\label\ltx@gobble
1913     \zref@extract{#1}{title}%
1914   \endgroup
1915 }%

```

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.

```

1916 \AtBeginDocument{%
1917   \ZREF@patch{@caption}{%

```

```

1918 \long\def\@caption#1[#2]{%
1919 \zref@titleref@setcurrent{#2}%
1920 \ZREF@org@caption{#1}[#2]}%
1921 }%
1922 }%

```

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

```

1923 \ZREF@patch{@part}{%
1924 \def\@part[#1]{%
1925 \zref@titleref@setcurrent{#1}%
1926 \ZREF@org@part[#{#1}]%
1927 }%
1928 }%
1929 \ZREF@patch{@chapter}{%
1930 \def\@chapter[#1]{%
1931 \zref@titleref@setcurrent{#1}%
1932 \ZREF@org@chapter[#{#1}]%
1933 }%
1934 }%
1935 \ZREF@patch{@sect}{%
1936 \def\@sect#1#2#3#4#5#6[#7]{%
1937 \zref@titleref@setcurrent{#7}%
1938 \ZREF@org@sect{#1}{#2}{#3}{#4}{#5}{#6}[#{#7}]%
1939 }%
1940 }%

```

The star versions of the section commands.

```

1941 \ZREF@patch{@spart}{%
1942 \def\@spart#1{%
1943 \zref@titleref@setcurrent{#1}%
1944 \ZREF@org@spart{#1}%
1945 }%
1946 }%
1947 \ZREF@patch{@schapter}{%
1948 \def\@schapter#1{%
1949 \zref@titleref@setcurrent{#1}%
1950 \ZREF@org@schapter{#1}%
1951 }%
1952 }%
1953 \ZREF@patch{@ssect}{%
1954 \def\@ssect#1#2#3#4#5{%
1955 \zref@titleref@setcurrent{#5}%
1956 \ZREF@org@ssect{#1}{#2}{#3}{#4}{#5}%
1957 }%
1958 }%

```

6.15.4 Environment description

```

1959 \ZREF@patch{descriptionlabel}{%
1960 \def\descriptionlabel#1{%
1961 \zref@titleref@setcurrent{#1}%
1962 \ZREF@org@descriptionlabel{#1}%
1963 }%
1964 }%

```

6.15.5 Class memoir

```

1965 \@ifclassloaded{memoir}{%
1966 \ltx@ifundefined{ifheadnameref}{-}{%
1967 \def\@chapter[#1]#2{%
1968 \ltx@ifundefined{choptc}{%
1969 \zref@titleref@setcurrent{#1}%
1970 }{-}%

```

```

1971     \ifx\ch@pt@c\ltx@empty
1972     \zref@titleref@setcurrent{#2}%
1973     \else
1974     \def\NR@temp{#1}%
1975     \ifx\NR@temp\ltx@empty
1976     \expandafter\zref@titleref@setcurrent
1977     \expandafter{\ch@pt@c}%
1978     \else
1979     \ifheadnameref
1980     \zref@titleref@setcurrent{#1}%
1981     \else
1982     \expandafter\zref@titleref@setcurrent
1983     \expandafter{\ch@pt@c}%
1984     \fi
1985     \fi
1986     \fi
1987     }%
1988     \ZREF@org@@chapter[#{#1}]{#2}%
1989     }%
1990     \ZREF@patch{M@sect}{%
1991     \def\M@sect#1#2#3#4#5#6[#7][#8]{%
1992     \ifheadnameref
1993     \zref@titleref@setcurrent{#8}%
1994     \else
1995     \zref@titleref@setcurrent{#7}%
1996     \fi
1997     \ZREF@org@M@sect{#1}{#2}{#3}{#4}{#5}{#6}[#{#7}][#{#8}]%
1998     }%
1999     }%
2000     }%
2001     }{}%

```

6.15.6 Class beamer

```

2002 \@ifclassloaded{beamer}{%
2003   \ZREF@patch{beamer@section}{%
2004     \long\def\beamer@section[#1]{%
2005       \zref@titleref@setcurrent{#1}%
2006       \ZREF@org@beamer@section[#{#1}]%
2007     }%
2008   }%
2009   \ZREF@patch{beamer@subsection}{%
2010     \long\def\beamer@subsection[#1]{%
2011       \zref@titleref@setcurrent{#1}%
2012       \ZREF@org@beamer@subsection[#{#1}]%
2013     }%
2014   }%
2015   \ZREF@patch{beamer@subsubsection}{%
2016     \long\def\beamer@subsubsection[#1]{%
2017       \zref@titleref@setcurrent{#1}%
2018       \ZREF@org@beamer@subsubsection[#{#1}]%
2019     }%
2020   }%
2021   }{}%

```

6.15.7 Package titlesec

```

2022 \@ifpackageloaded{titlesec}{%
2023   \ZREF@patch{ttl@sect@i}{%
2024     \def\ttl@sect@i#1#2[#3]#4{%
2025       \zref@titleref@setcurrent{#4}%
2026       \ZREF@org@ttl@sect@i{#1}{#2}[#{#3}]{#4}%
2027     }%
2028   }%
2029   \ZREF@patch{ttl@straight@i}{%

```

```

2030     \def\ttl@straight@i#1[#2]#3{%
2031         \def\ZREF@temp{#2}%
2032         \ifx\ZREF@temp\ltx@empty
2033             \zref@titleref@setcurrent{#3}%
2034         \else
2035             \zref@titleref@setcurrent{#2}%
2036         \fi
2037         \ZREF@org@ttl@straight@i{#1}[{#2}]{#3}%
2038     }%
2039 }%
2040 }{}%

```

6.15.8 Package longtable

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

```

2041 \@ifpackageloaded{longtable}{%
2042     \ZREF@patch{LT@caption}{%
2043         \def\LT@caption#1[#2]#3{%
2044             \ZREF@org@LT@caption{#1}[{#2}]{#3}%
2045             \zref@titleref@setcurrent{#2}%
2046         }%
2047     }%
2048 }{}%

```

6.15.9 Package listings

Package listings: support for its caption.

```

2049 \@ifpackageloaded{listings}{%
2050     \ZREF@patch{lst@MakeCaption}{%
2051         \def\lst@MakeCaption{%
2052             \ifx\lst@label\ltx@empty
2053             \else
2054                 \expandafter\zref@titleref@setcurrent\expandafter{%
2055                     \lst@caption
2056                 }%
2057             \fi
2058             \ZREF@org@lst@MakeCaption
2059         }%
2060     }%
2061 }{}%

```

6.15.10 Theorems

```

2062 \ZREF@patch{@opargbegintheorem}{%
2063     \def\@opargbegintheorem#1#2#3{%
2064         \zref@titleref@setcurrent{#3}%
2065         \ZREF@org@@opargbegintheorem{#1}{#2}{#3}%
2066     }%
2067 }%
2068 \@ifpackageloaded{amsthm}{%
2069     \begingroup
2070     \edef\x{macro:\string#1\string#2[\string#3]}%
2071     \@onelevel@sanitize\x
2072     \def\y#1->#2\@nil{#1}%
2073     \edef\z{\expandafter\y\meaning\@begintheorem->\@nil}%
2074     \@onelevel@sanitize\z
2075     \expandafter\endgroup
2076     \ifx\x\z
2077         \ZREF@patch{@begintheorem}{%
2078             \def\@begintheorem#1#2[#3]{%
2079                 \zref@titleref@setcurrent{#3}%
2080                 \ZREF@org@@begintheorem{#1}{#2}[{#3}]%

```

```

2081     }%
2082   }%
2083   \fi
2084 }{}%
2085 }
2086 </titleref>

```

6.16 Module xr

```

2087 <*xr>
2088 \NeedsTeXFormat{LaTeX2e}
2089 \ProvidesPackage{zref-xr}%
2090 [2026-03-07 v2.37 Module xr for zref (HO)]%
2091 \RequirePackage{zref-base}[2019/11/29]
2092 \ifx\ZREF@base@ok Y%
2093 \else
2094   \expandafter\endinput
2095 \fi
2096 \RequirePackage{keyval}
2097 \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.

```

2098 \zref@newprop{url}{}%
2099 \zref@newprop{urluse}{}%
2100 \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.

```

2101 \providecommand*{\zref@xr@ext}{}%
2102 \ltx@ifundefined{XR@ext}{pdf}{\XR@ext}%
2103 }%

```

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

```

2104 \newif\ifZREF@xr@zreflabel
2105 \SetupKeyvalOptions{%
2106   family=ZREF@XR,%
2107   prefix=ZREF@xr%
2108 }
2109 \DeclareBoolOption[true]{tozreflabel}
2110 \DeclareBoolOption[false]{toltxlabel}
2111 \DeclareBoolOption{verbose}
2112 \define@key{ZREF@XR}{ext}{}%
2113 \def\zref@xr@{#1}%
2114 }
2115 \DeclareBoolOption[false]{urluse}

```

`\zxrsetup`

```

2116 \newcommand*{\zxrsetup}{}%
2117 \kvsetkeys{ZREF@XR}%
2118 }%

```

`\ZREF@xr@URL`

```

2119 \newcount\ZREF@xr@URL
2120 \ZREF@xr@URL=\ltx@zero

```

`\ZREF@xr@AddURL`

```
2121 \def\ZREF@xr@AddURL#1{%
2122   \begingroup
2123     \def\ZREF@temp{#1}%
2124     \count@=\ltx@one
2125     \ZREF@xr@@AddUrl
2126   \endgroup
2127 }
```

`\ZREF@xr@@AddUrl`

```
2128 \def\ZREF@xr@@AddUrl{%
2129   \ifnum\count@>\ZREF@xr@URL
2130     \global\advance\ZREF@xr@URL by\ltx@one
2131     \xdef\ZREF@xr@theURL{\romannumeral\ZREF@xr@URL}%
2132     \global\expandafter\let
2133       \csname Z@U@\ZREF@xr@theURL\endcsname\ZREF@temp
2134     \@PackageInfo{zref-xr}{%
2135       \ltx@backslashchar Z@U@\ZREF@xr@theURL:\MessageBreak
2136       \ZREF@temp\MessageBreak
2137     }%
2138   \else
2139     \expandafter
2140     \ifx\csname Z@U@\romannumeral\count@\endcsname\ZREF@temp
2141       \xdef\ZREF@xr@theURL{\romannumeral\count@}%
2142     \else
2143       \expandafter\expandafter\expandafter\ZREF@xr@@AddUrl
2144     \fi
2145   \fi
2146 }
```

`\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).

```
2147 \ZREF@IfDefinable\zexternaldocument\def{%
2148   {%
2149     \ZREF@NewPropAnchor
2150     \ZREF@NewPropTitle
2151     \begingroup
2152     \csname @safe@actives@true\endcsname
2153     \makeatletter
2154     \@ifstar{%
2155       \ZREF@xr@zreflabelfalse
2156       \@testopt\ZREF@xr@externaldocument{}%
2157     }{%
2158       \ZREF@xr@zreflabeltrue
2159       \@testopt\ZREF@xr@externaldocument{}%
2160     }%
2161   }%
2162 }
```

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

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

```
2163 \def\ZREF@xr@externaldocument[#1]#2{%
2164   \def\ZREF@xr@prefix{#1}%
2165   \let\ZREF@xr@filelist\ltx@empty
2166   \edef\ZREF@xr@externalfile{#2}%
2167   \edef\ZREF@xr@file{\ZREF@xr@externalfile.aux}%
2168   \filename@parse{#2}%
```

```

2169     \@testopt\ZREF@xr@graburl{#2.\zref@xr@ext}%
2170 }%
2171 \def\ZREF@xr@graburl[#1]{%
2172     \edef\ZREF@xr@url{#1}%
2173     \ifZREF@xr@urluse
2174         \expandafter\ZREF@xr@AddURL\expandafter{\ZREF@xr@url}%
2175         \expandafter\def\expandafter\ZREF@xr@url
2176         \expandafter{\csname Z@U@\ZREF@xr@theURL\endcsname}%
2177     \fi
2178     \ZREF@xr@checkfile
2179 \endgroup
2180 }%

```

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

```

2181 \def\ZREF@xr@checkfile{%
2182     \openin\@inputcheck\ZREF@xr@file\relax
2183     \ifeof\@inputcheck
2184         \PackageWarning{zref-xr}{%
2185             File '\ZREF@xr@file' not found or empty,\MessageBreak
2186             labels not imported%
2187         }%
2188     \else
2189         \PackageInfo{zref-xr}{%
2190             Label \ifZREF@xr@zreflabel (zref) \fi
2191             import from '\ZREF@xr@file'%
2192         }%
2193         \def\ZREF@xr@found{0}%
2194         \def\ZREF@xr@ignored@empty{0}%
2195         \def\ZREF@xr@ignored@zref{0}%
2196         \def\ZREF@xr@ignored@ltx{0}%
2197         \ZREF@xr@processfile
2198         \closein\@inputcheck
2199         \begingroup
2200             \let\on@line\ltx@empty
2201             \PackageInfo{zref-xr}{%
2202                 Statistics for '\ZREF@xr@file':\MessageBreak
2203                 \ZREF@xr@found\space
2204                 \ifZREF@xr@zreflabel zref\else LaTeX\fi\space
2205                 label(s) found%
2206                 \ifnum\ZREF@xr@ignored@empty>0 %
2207                     ,\MessageBreak
2208                     \ZREF@xr@ignored@empty\space empty label(s) ignored%
2209                 \fi
2210                 \ifnum\ZREF@xr@ignored@zref>0 %
2211                     ,\MessageBreak
2212                     \ZREF@xr@ignored@zref\space
2213                     duplicated zref label(s) ignored%
2214                 \fi
2215                 \ifnum\ZREF@xr@ignored@ltx>0 %
2216                     ,\MessageBreak
2217                     \ZREF@xr@ignored@ltx\space
2218                     duplicated latex label(s) ignored%
2219                 \fi
2220             }%
2221         \endgroup
2222     \fi
2223     \ifx\ZREF@xr@filelist\ltx@empty
2224     \else
2225         \edef\ZREF@xr@file{%
2226             \expandafter\ltx@car\ZREF@xr@filelist\@nil
2227         }%
2228         \edef\ZREF@xr@filelist{%

```

```

2229     \expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil
2230   }%
2231   \expandafter\ZREF@xr@checkfile
2232 \fi
2233 }%

```

\ZREF@xr@processfile

```

2234 \def\ZREF@xr@processfile{%
2235   \read\@inputcheck to\ZREF@xr@line
2236   \expandafter\ZREF@xr@processline\ZREF@xr@line..\ZREF@nil
2237   \ifeof\@inputcheck
2238   \else
2239     \expandafter\ZREF@xr@processfile
2240   \fi
2241 }%

```

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

```

2242 \long\def\ZREF@xr@processline#1#2#3\ZREF@nil{%
2243   \def\x{#1}%
2244   \toks@{#2}%
2245   \ifZREF@xr@zreflabel
2246     \ifx\x\ZREF@xr@zref@newlabel
2247       \expandafter
2248         \ZREF@xr@process@zreflabel\ZREF@xr@line...\ZREF@nil
2249     \fi
2250   \else
2251     \ifx\x\ZREF@xr@newlabel
2252       \expandafter
2253         \ZREF@xr@process@label\ZREF@xr@line... []\ZREF@nil
2254     \fi
2255   \fi
2256   \ifx\x\ZREF@xr@@input
2257     \edef\ZREF@xr@filelist{%
2258       \etex@unexpanded\expandafter{\ZREF@xr@filelist}%
2259       {\filename@area\the\toks@}%
2260     }%
2261   \fi
2262 }%
2263 \def\ZREF@xr@process@zreflabel\zref@newlabel#1#2#3\ZREF@nil{%
2264   \edef\ZREF@xr@refname{Z@R@\ZREF@xr@prefix#1}%
2265   \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2266   \def\x{#2}%
2267   \edef\ZREF@xr@tempname{$temp$}%
2268   \edef\ZREF@xr@temprefname{Z@R@\ZREF@xr@tempname}%
2269   \let\ZREF@xr@list\x
2270   \ifx\ZREF@xr@list\ltx@empty
2271     \PackageWarningNoLine{zref-xr}{%
2272       Label ‘#1’ without properties ignored\MessageBreak
2273       in file ‘\ZREF@xr@file’%
2274     }%
2275     \edef\ZREF@xr@ignored@empty{%
2276       \the\numexpr\ZREF@xr@ignored@empty+1\relax
2277     }%
2278   \else
2279     \expandafter\ZREF@xr@checklist\x\ZREF@nil
2280     \expandafter\let\csname\ZREF@xr@temprefname\endcsname\x
2281     \expandafter\ltx@LocalAppendToMacro
2282     \csname\ZREF@xr@temprefname\expandafter\endcsname
2283     \expandafter{%
2284       \expandafter\externaldocument\expandafter{%
2285         \ZREF@xr@externalfile
2286       }%
2287     }%

```



```

2350 \def\ZREF@xr@newlabel{\newlabel}%
2351 \def\ZREF@xr@@input{\@input}%
2352 \def\ZREF@xr@relax{\relax}%

```

\ZREF@xr@tolabel

```

2353 \def\ZREF@xr@tolabel#1#2{%
2354   \ifZREF@xr@verbose
2355     \PackageInfo{zref-xr}{%
2356       Import to LaTeX label ‘#2’%
2357     }%
2358   \fi
2359   \zref@wrapper@unexpanded{%
2360     \expandafter\xdef\csname r@#2\endcsname{%
2361       {%
2362         \ltx@ifundefined{M@TitleReference}{%
2363           \ltx@ifundefined{TR@TitleReference}{%
2364             \zref@extractdefault{#1}{default}{}}%
2365           }{%
2366             \noexpand\TR@TitleReference
2367             {\zref@extractdefault{#1}{default}{}}%
2368             {\zref@extractdefault{#1}{title}{}}%
2369           }%
2370         }{%
2371           \noexpand\M@TitleReference
2372           {\zref@extractdefault{#1}{default}{}}%
2373           {\zref@extractdefault{#1}{title}{}}%
2374         }%
2375       }%
2376       {\zref@extractdefault{#1}{page}{}}%
2377     \ltx@ifpackageloaded{nameref}{%
2378       {\zref@extractdefault{#1}{title}{}}%
2379       {\zref@extractdefault{#1}{anchor}{}}%
2380     \zref@ifrefcontainsprop{#1}{urluse}{%
2381       {\zref@extractdefault{#1}{urluse}{}}%
2382     }{%
2383       {\zref@extractdefault{#1}{url}{}}%
2384     }%
2385   }{%
2386   }%
2387 }%
2388 }

```

\ZREF@xr@zref@ignorewarning

```

2389 \def\ZREF@xr@zref@ignorewarning#1{%
2390   \PackageWarningNoLine{zref-xr}{%
2391     Zref label ‘#1’ is already in use\MessageBreak
2392     in file ‘\ZREF@xr@file’%
2393   }%
2394   \edef\ZREF@xr@ignored@zref{%
2395     \the\numexpr\ZREF@xr@ignored@zref+1%
2396   }%
2397 }%

```

\ZREF@xr@ltx@ignorewarning

```

2398 \def\ZREF@xr@ltx@ignorewarning#1{%
2399   \PackageWarningNoLine{zref-xr}{%
2400     LaTeX label ‘#1’ is already in use\MessageBreak
2401     in file ‘\ZREF@xr@file’%
2402   }%
2403   \edef\ZREF@xr@ignored@ltx{%
2404     \the\numexpr\ZREF@xr@ignored@ltx+1%
2405   }%
2406 }%

```

\ZREF@xr@checklist

```
2407 \def\ZREF@xr@checklist#1#2#3\ZREF@nil{%
2408   \ifx\@undefined#1\relax
2409     \expandafter\ZREF@xr@checkkey\string#1\@nil
2410   \fi
2411   \ifx\#3\%
2412   \else
2413     \ltx@ReturnAfterFi{%
2414       \ZREF@xr@checklist#3\ZREF@nil
2415     }%
2416   \fi
2417 }%
2418 \def\ZREF@xr@checkkey#1#2\@nil{%
2419   \zref@ifpropundefined{#2}{%
2420     \zref@newprop{#2}{}%
2421   }{%
2422 }%
```

\ZREF@xr@scanparams

```
2423 \def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{%
2424   \let#1\ltx@empty
2425   \ZREF@foundfalse
2426   \ZREF@xr@scantitleref#1#2\TR@TitleReference{}-{}\ZREF@nil
2427   \ifZREF@found
2428   \else
2429     \ltx@LocalAppendToMacro#1{\default{#2}}%
2430   \fi
2431   % page
2432   \ltx@LocalAppendToMacro#1{\page{#3}}%
2433   % nameref title
2434   \ifZREF@found
2435   \else
2436     \ifx\#4\%
2437     \else
2438       \def\ZREF@xr@temp{#4}%
2439       \ifx\ZREF@xr@temp\ZREF@xr@relax
2440       \else
2441         \ltx@LocalAppendToMacro#1{\title{#4}}%
2442       \fi
2443     \fi
2444   \fi
2445   % anchor
2446   \ifx\#5\%
2447   \else
2448     \ltx@LocalAppendToMacro#1{\anchor{#5}}%
2449   \fi
2450   \ifx\#6\%
2451   \else
2452     \ifZREF@xr@urluse
2453       \ZREF@xr@AddURL{#6}%
2454       \expandafter\ltx@LocalAppendToMacro\expandafter#1%
2455       \expandafter{%
2456         \expandafter\urluse\expandafter{%
2457           \csname Z@U@\ZREF@xr@theURL\endcsname
2458         }%
2459       }%
2460     \else
2461       \ltx@LocalAppendToMacro#1{\url{#6}}%
2462     \fi
2463   \fi
2464 }%
```

`\ZREF@xr@scantitleref`

```
2465 \def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{%
2466   \ifx\#5\%
2467   \else
2468     \ltx@LocalAppendToMacro#1{%
2469       \default{#3}%
2470       \title{#4}%
2471     }%
2472     \ZREF@foundtrue
2473   \fi
2474 }%
```

`\ZREF@xr@urlcheck`

```
2475 \def\ZREF@xr@urlcheck#1{%
2476   \zref@ifrefcontainsprop{#1}{anchor}{%
2477     \zref@ifrefcontainsprop{#1}{url}{%
2478     }{%
2479     \expandafter
2480     \ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
2481     \expandafter{%
2482       \csname url\ifZREF@xr@urluse use\fi
2483       \expandafter\endcsname\expandafter{\ZREF@xr@url}%
2484     }%
2485   }%
2486 }{%
2487 }%
2488 }%
2489 \</xr>
```

6.17 Module `hyperref`

UNFINISHED :-(
(

```
2490 \*hyperref)
2491 \NeedsTeXFormat{LaTeX2e}
2492 \ProvidesPackage{zref-hyperref}%
2493 [2026-03-07 v2.37 Module hyperref for zref (HO)]%
2494 \RequirePackage{zref-base}[2019/11/29]
2495 \ifx\ZREF@base@ok Y%
2496 \else
2497   \expandafter\endinput
2498 \fi
2499 \ZREF@NewPropAnchor
2500 \zref@addprop\ZREF@mainlist{anchor}%
2501 \</hyperref)
```

6.18 Module `savepos`

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

6.18.1 Identification

```
2502 \*savepos)
2503 \NeedsTeXFormat{LaTeX2e}
2504 \ProvidesPackage{zref-savepos}%
2505 [2026-03-07 v2.37 Module savepos for zref (HO)]%
2506 \RequirePackage{zref-base}[2019/11/29]
2507 \ifx\ZREF@base@ok Y%
2508 \else
2509   \expandafter\endinput
```

2510 \fi

6.18.2 Availability

We now assume that it is available always.

6.18.3 Setup

```
2511 \zref@newlist{savepos}
2512 \ExplSyntaxOn
2513 \zref@newprop*{posx}[0]{\int_use:N \tex_lastxpos:D}
2514 \zref@newprop*{posy}[0]{\int_use:N \tex_lastypos:D}
2515 \zref@addprops{savepos}{posx,posy}
```

6.18.4 User macros

\zref@savepos

```
2516 \def\zref@savepos{%
2517   \if@filesw
2518     \tex_savepos:D
2519   \fi}
2520 \ExplSyntaxOff
```

\ZREF@zsavepos

```
2521 \def\ZREF@zsavepos#1#2#3{%
2522   \@bsphack
2523   \if@filesw
2524     \zref@savepos
2525     #1{#3}{#2}%
2526     \zref@savepos
2527   \fi
2528   \@esphack
2529 }
```

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

```
2530 \ZREF@IfDefinable\zsavpos\def{%
2531   {%
2532     \ZREF@zsavepos\zref@labelbylist{savepos}%
2533   }%
2534 }
```

\zsavposx

```
2535 \ZREF@IfDefinable\zsavposx\def{%
2536   {%
2537     \ZREF@zsavepos\zref@labelbyprops{posx}%
2538   }%
2539 }
```

\zsavposy

```
2540 \ZREF@IfDefinable\zsavposy\def{%
2541   {%
2542     \ZREF@zsavepos\zref@labelbyprops{posy}%
2543   }%
2544 }
```

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

```
2545 \newcommand*{\zposx}[1]{%
2546   \zref@extract{#1}{posx}%
2547 }%
2548 \newcommand*{\zposy}[1]{%
2549   \zref@extract{#1}{posy}%
2550 }%
```

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

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

`\ZREF@savepos@ok`

```
2551 \let\ZREF@savepos@ok=Y
```

```
2552 </savepos>
```

6.19 Module `abspos`

6.19.1 Identification

```
2553 <*abspos>
```

```
2554 \NeedsTeXFormat{LaTeX2e}
```

```
2555 \ProvidesPackage{zref-abspos}%
```

```
2556 [2026-03-07 v2.37 Module abspos for zref (H0)]%
```

```
2557 \RequirePackage{zref-base}[2019/11/29]
```

```
2558 \ifx\ZREF@base@ok Y%
```

```
2559 \else
```

```
2560 \expandafter\endinput
```

```
2561 \fi
```

```
2562 \RequirePackage{zref-savepos}[2019/11/29]
```

```
2563 \ifx\ZREF@savepos@ok Y%
```

```
2564 \else
```

```
2565 \expandafter\endinput
```

```
2566 \fi
```

```
2567 \RequirePackage{zref-pagelayout}[2019/11/29]
```

```
2568 \zref@addprop{savepos}{abspage}
```

```
2569 \zref@addprop{savepos}{pagevalue}
```

`\zref@absposx`

```
2570 \newcommand*{\zref@absposx}[3]{%
```

```
2571 \number
```

```
2572 \expandafter\zref@absposnumx\expandafter{%
```

```
2573 \number\zref@extractdefault{#1}{abspage}{0}%
```

```
2574 }{#2}{#3}%
```

```
2575 \ltx@space
```

```
2576 }
```

`\zref@absposy`

```
2577 \newcommand*{\zref@absposy}[3]{%
```

```
2578 \number
```

```
2579 \expandafter\zref@absposnumy\expandafter{%
```

```
2580 \number\zref@extractdefault{#1}{abspage}{0}%
```

```
2581 }{#2}{#3}%
```

```
2582 \ltx@space
```

```
2583 }
```

`\zref@absposnumx`

```
2584 \newcommand*{\zref@absposnumx}[3]{%
```

```
2585 \number
```

```
2586 \numexpr\ZREF@absposnum{thepage#1}{#2}{x}{#3}\relax
```

```
2587 }
```

`\zref@absposnumy`

```
2588 \newcommand*{\zref@absposnumy}[3]{%
```

```
2589 \number
```

```
2590 \numexpr\ZREF@absposnum{thepage#1}{#2}{y}{#3}\relax
```

```
2591 }
```

```

\ZREF@absposnum
2592 \def\ZREF@absposnum#1#2#3#4{%
2593 \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
2594 0%
2595 }{%
2596 \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
2597 }%
2598 }

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

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

\zref@def@absposnumx
2605 \ZREF@Robust\def\zref@def@absposnumx#1{%
2606 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumx}%
2607 }

\zref@def@absposnumy
2608 \ZREF@Robust\def\zref@def@absposnumy#1{%
2609 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumy}%
2610 }

\ZREF@def@abspos
2611 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2612 \edef#1{#2{#3}{#4}{#5}}%
2613 }

\zref@absposused
2614 \ZREF@Robust\def\zref@absposused{%
2615 \zref@wrapper@babel\ZREF@abspos@used
2616 }

\ZREF@abspos@used
2617 \def\ZREF@abspos@used#1{%
2618 \zref@refused{#1}%
2619 \zref@ifrefundefined{#1}{%
2620 }{%
2621 \begingroup
2622 \edef\ZREF@temp{%
2623 \zref@extractdefault{#1}{abspage}{0}%
2624 }%
2625 \ifnum\ZREF@temp>\ltx@zero
2626 \zref@refused{thepage\ZREF@temp}%
2627 \else
2628 \@PackageError{zref-abspos}{%
2629 \string\zref@pos@label@used\ltx@space
2630 needs property 'abspage'\MessageBreak
2631 in label '#1'%
2632 }\@ehc
2633 \fi
2634 \endgroup
2635 }%
2636 }

```

```

\zref@absposnumused
2637 \newcommand*\zref@absposnumused}[1]{%
2638   \ifnum#1>\ltx@zero
2639     \zref@refused{thepage\number#1}%
2640   \else
2641     \@PackageError{zref-abspos}{%
2642       Invalid absolute page number (#1)\MessageBreak
2643       for \string\zref@pos@num@used.\MessageBreak
2644       A positive integer is expected%
2645     }\@ehc
2646   \fi
2647 }

```

```

\zref@ifabsposundefined
2648 \def\zref@ifabsposundefined#1{%
2649   \zref@ifrefundefined{#1}\ltx@firsttwo{%
2650     \expandafter\zref@ifabsposnumundefined\expandafter{%
2651       \number\zref@extractdefault{#1}{abspage}{0}%
2652     }%
2653   }%
2654 }

```

```

\zref@ifabsposnumundefined
2655 \def\zref@ifabsposnumundefined#1{%
2656   \ifnum\ZREF@number{#1}>\ltx@zero
2657     \zref@ifrefundefined{thepage#1}%
2658     \ltx@firstoftwo\ltx@secondoftwo
2659   \else
2660     \expandafter\ltx@firstoftwo
2661   \fi
2662 }

```

6.19.2 Media

Ensure that `\stockwidth` and `\stockheight` are defined

```

2663 \@ifundefined{stockwidth}{\newdimen\stockwidth}{}
2664 \@ifundefined{stockheight}{\newdimen\stockheight}{}

```

```

\ZREF@abspos@media@width
2665 \edef\ZREF@abspos@media@width{%
2666   \ltx@ifundefined{pdfpagewidth}{%
2667     \ltx@ifundefined{mediawidth}{%
2668       \ifdim\stockwidth > \z@
2669         stockwidth%
2670       \else
2671         paperwidth%
2672       \fi
2673     }{%
2674       mediawidth%
2675     }%
2676   }{%
2677     pdfpagewidth%
2678   }%
2679 }
2680 \ifluatex
2681 \def\ZREF@abspos@media@width{pdfpagewidth}%
2682 \fi

```

```

\ZREF@abspos@media@height
2683 \edef\ZREF@abspos@media@height{%
2684   \ltx@ifundefined{pdfpageheight}{%
2685     \ltx@ifundefined{mediaheight}{%

```

```

2686     \ifdim\stockwidth > \z@
2687         stockheight%
2688     \else
2689         paperheight%
2690     \fi
2691 }{%
2692     mediaheight%
2693 }%
2694 }{%
2695     \noexpand\ifcase\pdfpageheight
2696     \ifdim\stockwidth > \z@
2697         stockheight%
2698     \else
2699         paperheight%
2700     \fi
2701     \noexpand\else
2702     pdfpageheight%
2703     \noexpand\fi
2704 }%
2705 }
2706 \ifluatex
2707 \edef\ZREF@abspos@media@height{%
2708     \noexpand\ifcase\pageheight
2709     \ifdim\stockwidth > \z@
2710         stockheight%
2711     \else
2712         paperheight%
2713     \fi
2714     \noexpand\else
2715     pdfpageheight%
2716     \noexpand\fi}%
2717 \fi

```

\ZREF@abspos@media@x@left

```

2718 \def\ZREF@abspos@media@x@left#1{%
2719     0%
2720 }

```

\ZREF@abspos@media@x@right

```

2721 \def\ZREF@abspos@media@x@right#1{%
2722     \zref@extract{#1}\ZREF@abspos@media@width
2723 }

```

\ZREF@abspos@media@x@center

```

2724 \def\ZREF@abspos@media@x@center#1{%
2725     \ZREF@abspos@media@x@left{#1}%
2726     +\zref@extract{#1}\ZREF@abspos@media@width/2%
2727 }

```

\ZREF@abspos@media@y@top

```

2728 \def\ZREF@abspos@media@y@top#1{%
2729     \zref@extract{#1}\ZREF@abspos@media@height
2730 }

```

\ZREF@abspos@media@y@bottom

```

2731 \def\ZREF@abspos@media@y@bottom#1{%
2732     0%
2733 }

```

\ZREF@abspos@media@y@center

```

2734 \def\ZREF@abspos@media@y@center#1{%
2735     \zref@extract{#1}\ZREF@abspos@media@height/2%
2736 }

```

6.19.3 Paper

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

```
2737 \def\ZREF@abspos@paper@x@left#1{%
2738   0%
2739 }
```

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

```
2740 \def\ZREF@abspos@paper@x@right#1{%
2741   \zref@extract{#1}{paperwidth}%
2742 }
```

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

```
2743 \def\ZREF@abspos@paper@x@center#1{%
2744   \zref@extract{#1}{paperwidth}/2%
2745 }
```

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

```
2746 \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top
```

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

```
2747 \def\ZREF@abspos@paper@y@bottom#1{%
2748   \ZREF@abspos@paper@y@top{#1}%
2749   -\zref@extract{#1}{paperheight}%
2750 }
```

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

```
2751 \def\ZREF@abspos@paper@y@center#1{%
2752   \ZREF@abspos@paper@y@top{#1}%
2753   -\zref@extract{#1}{paperheight}/2%
2754 }
```

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`

```
2755 \let\ZREF@temp\ltx@two
2756 \ltx@ifundefined{pdfhorigin}{}%
2757 % \ifpdf
2758   \let\ZREF@temp\ltx@zero
2759 % \fi
2760 }
2761 \ifluatex
2762 % \ifpdf
2763   \let\ZREF@temp\ltx@zero
2764 % \fi
2765 \fi
2766
2767 \ifx\ZREF@temp\ltx@two
2768   \ifnum\mag=1000 %
2769     \let\ZREF@temp\ltx@one
2770   \fi
2771 \fi
2772 \ifcase\ZREF@temp
2773   \def\ZREF@abspos@origin@x#1{%
2774     \zref@extract{#1}{pdfhorigin}%
2775   }%
2776 \or
2777   \def\ZREF@abspos@origin@x#1{%
2778     4736286%
```

```

2779 }%
2780 \or
2781 \def\ZREF@abspos@origin@x#1{%
2782 \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2783 }%
2784 \fi

```

\ZREF@abspos@origin@y

```

2785 \let\ZREF@temp\ltx@two
2786 \ltx@ifundefined{pdfvorigin}{}%
2787 % \ifpdf
2788 \let\ZREF@temp\ltx@zero
2789 % \fi
2790 }
2791 \ifluatex
2792 % \ifpdf
2793 \let\ZREF@temp\ltx@zero
2794 % \fi
2795 \fi
2796 \ifx\ZREF@temp\ltx@two
2797 \ifnum\mag=1000 %
2798 \let\ZREF@temp\ltx@one
2799 \fi
2800 \fi
2801 \ifcase\ZREF@temp
2802 \def\ZREF@abspos@origin@y#1{%
2803 \zref@extract{#1}{pdfvorigin}%
2804 }%
2805 \or
2806 \def\ZREF@abspos@origin@y#1{%
2807 4736286%
2808 }%
2809 \or
2810 \def\ZREF@abspos@origin@y#1{%
2811 \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2812 }%
2813 \fi

```

6.19.5 Header

\ZREF@abspos@head@x@left

```

2814 \def\ZREF@abspos@head@x@left#1{%
2815 \ZREF@abspos@paper@x@left{#1}%
2816 +\ZREF@abspos@origin@x{#1}%
2817 +\zref@extract{#1}{hoffset}%
2818 +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2819 \zref@extract{#1}{oddsidemargin}%
2820 \else
2821 \zref@extract{#1}{evensidemargin}%
2822 \fi
2823 }

```

\ZREF@abspos@head@x@right

```

2824 \def\ZREF@abspos@head@x@right#1{%
2825 \ZREF@abspos@head@x@left{#1}%
2826 +\zref@extract{#1}{textwidth}%
2827 }

```

\ZREF@abspos@head@x@center

```

2828 \def\ZREF@abspos@head@x@center#1{%
2829 \ZREF@abspos@head@x@left{#1}%
2830 +\zref@extract{#1}{textwidth}/2%
2831 }

```

```

\ZREF@abspos@head@y@top
2832 \def\ZREF@abspos@head@y@top#1{%
2833 \ZREF@abspos@paper@y@top{#1}%
2834 -\ZREF@abspos@origin@y{#1}%
2835 -\zref@extract{#1}{voffset}%
2836 -\zref@extract{#1}{topmargin}%
2837 }

```

```

\ZREF@abspos@head@y@bottom
2838 \def\ZREF@abspos@head@y@bottom#1{%
2839 \ZREF@abspos@head@y@top{#1}%
2840 -\zref@extract{#1}{headheight}%
2841 }

```

```

\ZREF@abspos@head@y@center
2842 \def\ZREF@abspos@head@y@center#1{%
2843 \ZREF@abspos@head@y@top{#1}%
2844 -\zref@extract{#1}{headheight}/2%
2845 }

```

6.19.6 Body

```

\ZREF@abspos@body@x@left
2846 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left

```

```

\ZREF@abspos@body@x@right
2847 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right

```

```

\ZREF@abspos@body@x@center
2848 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center

```

```

\ZREF@abspos@body@y@top
2849 \def\ZREF@abspos@body@y@top#1{%
2850 \ZREF@abspos@head@y@bottom{#1}%
2851 -\zref@extract{#1}{headsep}%
2852 }

```

```

\ZREF@abspos@body@y@bottom
2853 \def\ZREF@abspos@body@y@bottom#1{%
2854 \ZREF@abspos@body@y@top{#1}%
2855 -\zref@extract{#1}{textheight}%
2856 }

```

```

\ZREF@abspos@body@y@center
2857 \def\ZREF@abspos@body@y@center#1{%
2858 \ZREF@abspos@body@y@top{#1}%
2859 -\zref@extract{#1}{textheight}/2%
2860 }

```

6.19.7 Footer

```

\ZREF@abspos@foot@x@left
2861 \let\ZREF@abspos@foot@x@left\ZREF@abspos@head@x@left

```

```

\ZREF@abspos@foot@x@right
2862 \let\ZREF@abspos@foot@x@right\ZREF@abspos@head@x@right

```

```

\ZREF@abspos@foot@x@center
2863 \let\ZREF@abspos@foot@x@center\ZREF@abspos@head@x@center

```

```

\ZREF@abspos@foot@y@bottom
2864 \def\ZREF@abspos@foot@y@bottom#1{%
2865 \ZREF@abspos@body@y@bottom{#1}%
2866 -\zref@extract{#1}{footskip}%
2867 }

```

6.19.8 Marginal notes

```

\ZREF@abspos@marginpar@x@left
2868 \def\ZREF@abspos@marginpar@x@left#1{%
2869 \ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2870 \ZREF@abspos@body@x@right{#1}%
2871 +\zref@extract{#1}{marginparsep}%
2872 \else
2873 \ZREF@abspos@body@x@left{#1}%
2874 -\zref@extract{#1}{marginparsep}%
2875 -\zref@extract{#1}{marginparwidth}%
2876 \fi
2877 }

```

```

\ZREF@abspos@marginpar@x@right
2878 \def\ZREF@abspos@marginpar@x@right#1{%
2879 \ZREF@abspos@marginpar@x@left{#1}%
2880 +\zref@extract{#1}{marginparwidth}%
2881 }

```

```

\ZREF@abspos@marginpar@x@center
2882 \def\ZREF@abspos@marginpar@x@center#1{%
2883 \ZREF@abspos@marginpar@x@left{#1}%
2884 +\zref@extract{#1}{marginparwidth}/2%
2885 }

```

```

\ZREF@abspos@marginpar@y@top
2886 \let\ZREF@abspos@marginpar@y@top\ZREF@abspos@body@y@top

```

```

\ZREF@abspos@marginpar@y@bottom
2887 \let\ZREF@abspos@marginpar@y@bottom\ZREF@abspos@body@y@bottom

```

```

\ZREF@abspos@marginpar@y@center
2888 \let\ZREF@abspos@marginpar@y@center\ZREF@abspos@body@y@center

```

6.19.9 Stock paper

```

\ZREF@abspos@stock@x@left
2889 \let\ZREF@abspos@stock@x@left\ZREF@abspos@paper@x@left

```

```

\ZREF@abspos@stock@x@right
2890 \let\ZREF@abspos@stock@x@right\ZREF@abspos@paper@x@right

```

```

\ZREF@abspos@stock@x@center
2891 \let\ZREF@abspos@stock@x@center\ZREF@abspos@paper@x@center

```

```

\ZREF@abspos@stock@y@top
2892 \let\ZREF@abspos@stock@y@top\ZREF@abspos@paper@y@top

```

```

\ZREF@abspos@stock@y@bottom
2893 \let\ZREF@abspos@stock@y@bottom\ZREF@abspos@paper@y@bottom

```

```

\ZREF@abspos@stock@y@center
2894 \let\ZREF@abspos@stock@y@center\ZREF@abspos@paper@y@center

```

```

2895 </abspos>

```

6.20 Module dotfill

```
2896 <*dotfill>
2897 \NeedsTeXFormat{LaTeX2e}
2898 \ProvidesPackage{zref-dotfill}%
2899 [2026-03-07 v2.37 Module dotfill for zref (HO)]%
2900 \RequirePackage{zref-base}[2019/11/29]
2901 \ifx\ZREF@base@ok Y%
2902 \else
2903 \expandafter\endinput
2904 \fi
```

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

```
2905 \RequirePackage{zref-savepos}[2019/11/29]
```

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

```
2906 \zref@require@unique
```

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

```
2907 \RequirePackage{keyval}
```

The definitions of the keys follow.

```
2908 \define@key{ZREF@DF}{unit}{%
```

```
2909 \def\ZREF@df@unit{#1}%
```

```
2910 }
```

```
2911 \define@key{ZREF@DF}{min}{%
```

```
2912 \def\ZREF@df@min{#1}%
```

```
2913 }
```

```
2914 \define@key{ZREF@DF}{dot}{%
```

```
2915 \def\ZREF@df@dot{#1}%
```

```
2916 }
```

Defaults are set, see user interface.

```
2917 \providecommand\ZREF@df@min{2}
```

```
2918 \providecommand\ZREF@df@unit{.44em}
```

```
2919 \providecommand\ZREF@df@dot{.}
```

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

```
2920 \newcommand*{\zdotfillsetup}{\kvsetkeys{ZREF@DF}}
```

`\zdotfill` `\zdotfill` sets labels at the left and the right to get the horizontal position.

`\zsavespos` is not used, because we do not need the vertical position.

```
2921 \ZREF@IfDefinable\zdotfill\def{%
```

```
2922 {%
```

```
2923 \leavevmode
```

```
2924 \global\advance\c@zref@unique\ltx@one
```

```
2925 \begingroup
```

```
2926 \def\ZREF@temp{zref@\number\c@zref@unique}%
```

```
2927 \csname tex_savepos:D\endcsname
```

```
2928 \zref@labelbyprops{\thezref@unique L}{posx}%
```

```
2929 \setlength{\dimen@}{\ZREF@df@unit}%
```

```
2930 \zref@ifrefundefined{\thezref@unique R}{%
```

```
2931 \ZREF@dotfill
```

```
2932 }{%
```

```
2933 \ifnum\numexpr\zposx{\thezref@unique R}%
```

```
2934 -\zposx{\thezref@unique L}\relax
```

```
2935 <\dimexpr\ZREF@df@min\dimen@\relax
```

```
2936 \hfill
```

```
2937 \else
```

```
2938 \ZREF@dotfill
```

```
2939 \fi
```

```
2940 }{%
```

```
2941 \csname tex_savepos:D\endcsname
```

```
2942 \zref@labelbyprops{\thezref@unique R}{posx}%
```

```

2943   \endgroup
2944   \kern\z@
2945 }%
2946 }

```

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

```

2947 \def\ZREF@dotfill{%
2948   \cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill
2949 }

2950 </dotfill>

```

6.21 Module `env`

```

2951 <*env>
2952 \NeedsTeXFormat{LaTeX2e}
2953 \ProvidesPackage{zref-env}%
2954   [2026-03-07 v2.37 Module env for zref (HO)]%
2955 \RequirePackage{zref-base}[2019/11/29]
2956 \ifx\ZREF@base@ok Y%
2957 \else
2958   \expandafter\endinput
2959 \fi

2960 \zref@newprop{envname}[]{\@currentenv}
2961 \zref@newprop{envline}[]{\zref@env@line}

```

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

```

2962 \def\zref@env@line{%
2963   \ifx\@currentline\ltx@empty
2964   \else
2965     \expandafter
2966     \ZREF@ENV@line\@currentline\ltx@empty line \ltx@empty\@nil
2967   \fi
2968 }

```

`\ZREF@ENV@line`

```

2969 \def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%
2970 </env>

```

7 Installation

7.1 Download

Package. This package is available on CTAN²:

[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 T_EX 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`):

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

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

7.4 Refresh file name databases

If your `TEX` distribution (`TEX Live`, `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 pdfL^AT_EX:

```
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 Arsenaу, 2001/04/05 v3.1. [CTAN:pkg/titleref](#)
- [7] Package `totpages`, Wilhelm Müller, 1999/07/14 v1.00. [CTAN:pkg/totpages](#)
- [8] Package `xr`, David Carlisle, 1994/05/28 v5.02. [CTAN:pkg/xr](#)
- [9] Package `xr-hyper`, David Carlisle, 2000/03/22 v6.00beta4. [CTAN:pkg/xr-hyper](#)

9 History

[2006/02/20 v1.0]

- First version.

[2006/05/03 v1.1]

- Module `perpage` added.
- Module redesign as packages.

[2006/05/25 v1.2]

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

[2006/09/08 v1.3]

- Typo fixes and English cleanup by Per Starback.

[2007/01/23 v1.4]

- Typo in macro name fixed in documentation.

[2007/02/18 v1.5]

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

[2007/04/06 v1.6]

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

[2007/04/17 v1.7]

- Package `atbegshi` replaces `everyshi`.

[2007/04/22 v1.8]

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

[2007/05/02 v1.9]

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

[2007/05/06 v2.0]

- Uses package `etexcmds` for accessing \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 luatex versions.
- added some documentation of the abspos module.
- adapted the abspos module to the new luatex primitives.
- adapted pageattr module to the new luatex primitives.
- added short documentation for pageattr module
- use luatex command names directly in zref-savepos rather than defining pdftex compatibility names.
- allow zref-abspos to use `\pdf[vh]origin` in dvi mode.

[2020-03-04 v2.30]

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

[2020-05-28 v2.31]

- Adapted module zref-counter to use `\@currentcounter` in the next L^AT_EX version (issue 5)

[2020-07-03 v2.32]

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

[2022-03-08 v2.33]

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

[2022-04-07 v2.34]

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

[2023-09-14 v.2.35]

- Define the 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>

- Updated `\zlabel` definition to new `\label` definition.
<https://github.com/ho-tex/zref/issues/23>

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	
<code>\@PackageError</code>	509, 525, 2628, 2641
<code>\@PackageInfo</code>	2134
<code>\@PackageInfoNoLine</code>	549, 564, 1380, 1475, 1487, 1554, 1606
<code>\@PackageWarning</code>	692
<code>\@addtoreset</code>	1026
<code>\@auxout</code>	716
<code>\@begintheorem</code>	2073, 2078
<code>\@bsphack</code>	601, 611, 631, 2522
<code>\@caption</code>	1918
<code>\@chapter</code>	1930, 1967
<code>\@currentHref</code>	945
<code>\@currentcounter</code>	1048
<code>\@currentlabel</code>	940
<code>\@currenvir</code>	2960
<code>\@currentline</code>	2963, 2966
<code>\@ehc</code>	297, 307, 492, 515, 527, 1638, 2632, 2645
<code>\@elt</code>	1761
<code>\@esphack</code>	608, 628, 643, 2528
<code>\@firstofone</code>	1594
<code>\@firstoftwo</code>	1460, 1538
<code>\@gobble@om</code>	976
<code>\@ifclassloaded</code>	1965, 2002
<code>\@ifdefinable</code>	243, 290
<code>\@ifl@t@r</code>	223, 1046, 1075
<code>\@ifnextchar</code>	531, 1784
<code>\@ifpackageloaded</code>	2022, 2041, 2049, 2068
<code>\@ifstar</code>	496, 2154
<code>\@ifundefined</code>	192, 910, 1024, 1795, 2290, 2332, 2663, 2664
<code>\@input</code>	2351
<code>\@inputcheck</code>	2182, 2183, 2198, 2235, 2237
<code>\@latex@warning</code>	778
<code>\@mainaux</code>	1723
<code>\@namedef</code>	536
<code>\@ne</code>	1760
<code>\@newl@bel</code>	286
<code>\@nil</code>	1653, 1655, 1883, 1890, 2072, 2073, 2226, 2229, 2409, 2418, 2966, 2969
<code>\@onelevel@sanitize</code>	423, 441, 506, 534, 2071, 2074
<code>\@opargbegintheorem</code>	2063
<code>\@part</code>	1924
<code>\@schapter</code>	1948
<code>\@secondoftwo</code>	1462, 1540
<code>\@sect</code>	1936
<code>\@spart</code>	1942
<code>\@ssect</code>	1954
<code>\@stpelt</code>	1761, 1771, 1772
<code>\@testdef</code>	1362, 1363, 1522, 1523
<code>\@testopt</code>	2156, 2159, 2169
<code>\@tfor</code>	323, 721
<code>\@undefined</code>	1839, 2408
<code>\@</code>	25, 26, 27, 28, 153, 155, 157, 158, 170, 173, 2317, 2411, 2436, 2446, 2450, 2466
<code>_</code>	44, 45
A	
<code>\AddLineBeginAux</code>	281
<code>\addtocounter</code>	1760
<code>\AddToHook</code>	917, 1031, 1076, 1087, 1134, 1371, 1444, 1531, 1615, 1754
<code>\advance</code>	1090, 1427, 1583, 1808, 2130, 2924
<code>\afterassignment</code>	234, 1174, 1178
<code>\Alph</code>	7
<code>\anchor</code>	2448
<code>\AtBeginDocument</code>	1054, 1263, 1717, 1916
<code>\AtBeginShipoutAddToBox</code>	1135
<code>\AtBeginShipoutBoxDepth</code>	1351
<code>\AtBeginShipoutBoxHeight</code>	1350
<code>\AtBeginShipoutBoxWidth</code>	1349
<code>\AtEndOfPackage</code>	195
B	
<code>\beamer@section</code>	2004
<code>\beamer@subsection</code>	2010
<code>\beamer@subsubsection</code>	2016
<code>\begin</code>	23, 57, 100, 106, 156, 172, 1770
<code>\bfseries</code>	933
C	
<code>\c@abspage</code>	1816, 1823
<code>\c@page</code>	959, 1090, 1813, 2818, 2869
<code>\c@zpage</code>	1813, 1819
<code>\c@zref@unique</code>	913, 920, 1808, 2924, 2926
<code>\ch@pt@c</code>	1971, 1977, 1983
<code>\chapter</code>	24, 30, 32, 61, 82
<code>\ChapterPages</code>	91, 112
<code>\ChapterStart</code>	78, 135, 150, 166
<code>\ChapterStop</code>	85, 148, 165, 184
<code>\chardef</code>	1196, 1211, 1220, 1224
<code>\cleaders</code>	2948
<code>\cleardoublepage</code>	79, 86

<code>\clearpage</code>	62	1829, 1831, 1832, 1839, 1892,
<code>\closein</code>	2198	1895, 2133, 2140, 2152, 2176,
<code>\columnsep</code>	1335	2280, 2282, 2297, 2298, 2315,
<code>\columnwidth</code>	1334	2322, 2325, 2339, 2340, 2360,
<code>\comma@entry</code>	344, 345, 347, 353, 379, 380, 382, 388, 615, 617, 621, 1659, 1660, 1661, 1667, 1670	2457, 2480, 2483, 2596, 2927, 2941
<code>\comma@parse</code>	343, 378, 614, 1658	<code>\endinput</code> 192, 265, 278, 971, 1021, 1044, 1072, 1125, 1165, 1256, 1274, 1455, 1500, 1626, 1737, 1850, 2094, 2497, 2509, 2560, 2565, 2903, 2958
<code>\count@</code>	1377, 1388, 1389, 1391, 1426, 1427, 1436, 1438, 1439, 1551, 1574, 1576, 1577, 1582, 1583, 2124, 2129, 2140, 2141	<code>\escapechar</code> 330, 421, 461, 462, 468, 723, 1280
<code>\cs</code>	1768	<code>\etex@unexpanded</code> 592, 820, 840, 2258
<code>\csname</code>	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, 1367, 1527, 1688, 1696, 1748, 1758, 1761, 1796, 1798, 1801, 1803, 1815, 1821, 1826, 1827, 1829, 1831, 1832, 1839, 1892, 1895, 2133, 2140, 2152, 2176, 2280, 2282, 2297, 2298, 2315, 2322, 2325, 2339, 2340, 2360, 2457, 2480, 2482, 2596, 2927, 2941	<code>\evensidemargin</code> 1326 <code>\ExplSyntaxOff</code> 2520 <code>\ExplSyntaxOn</code> 2512 <code>\externaldocument</code> 2284, 2326
<code>\current@chapid</code>	80, 88	
D		
<code>\DeclareBoolOption</code>	2109, 2110, 2111, 2115	
<code>\DeclareOption</code>	194	
<code>\default</code>	2429, 2469	
<code>\define@key</code>	1891, 1894, 1897, 1900, 2112, 2908, 2911, 2914	
<code>\descriptionlabel</code>	1960	
<code>\detokenize</code>	1878	
<code>\dftest</code>	167, 174, 175, 176, 177, 178, 179, 180, 181, 182	
<code>\dimen@</code>	2929, 2935, 2948	
<code>\dimexpr</code>	153, 155, 1439, 2782, 2811, 2935	
<code>\do</code>	328, 721	
<code>\documentclass</code>	4, 39, 68, 273	
<code>\dotfill</code>	169, 173	
E		
<code>\emph</code>	150	
<code>\end</code>	34, 64, 130, 159, 183, 185, 1765	
<code>\endcsname</code>	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, 1367, 1527, 1688, 1696, 1747, 1748, 1757, 1758, 1761, 1773, 1796, 1798, 1801, 1802, 1803, 1815, 1821, 1826, 1827,	
F		
<code>\fancyhead</code>	51, 54	
<code>\fancyhf</code>	50, 53	
<code>\fancypagestyle</code>	52	
<code>\filename@area</code>	2259	
<code>\filename@parse</code>	2168	
<code>\fmtversion</code>	223, 1046, 1075	
<code>\foo</code>	18, 29, 31, 33	
<code>\footskip</code>	1331	
<code>\foremargin</code>	1338	
<code>\frontmatter</code>	58, 103	
G		
<code>\g@addto@macro</code>	351, 369, 1801	
<code>\G@refundefinedtrue</code>	777	
<code>\gdef</code>	413, 540, 545, 950, 1367, 1724, 1796, 1798	
<code>\GetTitleStringDisableCommands</code>	1887	
<code>\GetTitleStringExpand</code>	1873	
<code>\GetTitleStringNonExpand</code>	1875	
<code>\GetTitleStringResult</code>	1878	
H		
<code>\hb@xt@</code>	2948	
<code>\headheight</code>	1329	
<code>\headmargin</code>	1341	
<code>\headsep</code>	1330	
<code>\hfill</code>	2936, 2948	
<code>\hoffset</code>	1322	
<code>\hss</code>	2948	
I		
<code>\if@filesw</code>	711, 1077, 1088, 1722, 2517, 2523	
<code>\if@safe@actives</code>	890	
<code>\ifcase</code>	115, 1232, 1659, 2695, 2708, 2772, 2801	
<code>\ifcsname</code>	884, 1747, 1757, 1773, 1802	
<code>\ifdim</code>	2668, 2686, 2696, 2709	
<code>\ifeof</code>	2183, 2237	
<code>\ifetex@unexpanded</code>	268	
<code>\iffirstchoice@</code>	1756	
<code>\IfFormatAtLeastTF</code>	223, 915, 973, 1342, 1752	
<code>\ifheadnameref</code>	1979, 1992	
<code>\ifin@</code>	318	

<code>\ifluatex</code>	. 1299, 1459, 1536, 1549, 1592, 1608, 2680, 2706, 2761, 2791	2032, 2052, 2165, 2200, 2223, 2229, 2270, 2424, 2963, 2966, 2969
<code>\ifnum</code> 477, 1102, 1207, 1217, 1223, 1426, 1582, 1631, 1684, 2129, 2206, 2210, 2215, 2625, 2638, 2656, 2768, 2797, 2933	<code>\ltx@firstofone</code>
<code>\ifodd</code> 124, 2818, 2869	... 255, 868, 879, 885, 1508, 1509
<code>\ifpdf</code> 2757, 2762, 2787, 2792	<code>\ltx@firstoftwo</code>
<code>\ifx</code> 438, 442, 475, 508, 566, 674, 677, 691, 730, 796, 969, 976, 985, 993, 1019, 1042, 1070, 1123, 1163, 1254, 1272, 1366, 1453, 1499, 1507, 1526, 1537, 1593, 1624, 1641, 1650, 1654, 1659, 1660, 1661, 1735, 1826, 1848, 1971, 1975, 2032, 2052, 2076, 2092, 2140, 2223, 2246, 2251, 2256, 2270, 2317, 2408, 2411, 2436, 2439, 2446, 2450, 2466, 2495, 2507, 2558, 2563, 2767, 2796, 2901, 2956, 2963	800, 827, 828, 893, 1104, 2658, 2660
<code>\ifZREF@found</code> 248, 2427, 2434	<code>\ltx@firsttwo</code>
<code>\ifZREF@immediate</code>	2649
 635, 701, 713, 717, 732	<code>\ltx@gobble</code>
<code>\ifZREF@pa@list</code> 1514, 1519	251,
<code>\ifZREF@pl@list</code> 1354, 1359	356, 391, 624, 664, 977, 985, 986, 993, 1429, 1585, 1672, 1912
<code>\ifzref@titleref@expand</code> 1856, 1872	<code>\ltx@gobblethree</code>
<code>\ifzref@titleref@stripperiod</code>	994
 1869, 1880	<code>\ltx@gobbletwo</code> .
<code>\ifZREF@xrr@toltxlabel</code>	... 2303, 2345	695, 1026, 1674, 1778
<code>\ifZREF@xrr@tozreflabel</code>	.. 2289, 2331	<code>\ltx@ifpackageloaded</code>
<code>\ifZREF@xrr@urluse</code>	.. 2173, 2452, 2482	2377
<code>\ifZREF@xrr@verbose</code>	.. 2291, 2333, 2354	<code>\ltx@ifUndefined</code>
<code>\ifZREF@xrr@zreflabel</code> 230, 250, 258, 411, 878, 924, 1140, 1474, 1486, 1505, 1506, 1535, 1591, 1966, 1968, 2756, 2786
 2104, 2190, 2204, 2245	<code>\ltx@ifundefined</code>
<code>\immediate</code> 706, 1723	. 301, 486, 584, 761, 806, 945, 1281, 1361, 1521, 2102, 2362, 2363, 2593, 2666, 2667, 2684, 2685
<code>\in@</code> 315	<code>\ltx@LocalAppendToMacro</code>
<code>\int</code> 2513, 2514 386, 404, 650, 660, 1605, 2281, 2321, 2324, 2429, 2432, 2441, 2448, 2454, 2461, 2468, 2480
<code>\item</code> 107, 111, 113, 121, 125, 127	<code>\ltx@newif</code>

K

<code>\kern</code> 2944
<code>\kv@define@key</code>	645, 656, 667, 672, 689
<code>\kv@key</code>	... 693, 1653, 1655, 1656, 1670
<code>\kv@parse</code> 690, 1649
<code>\kv@value</code> 691, 1650, 1651, 1658
<code>\kvsetkeys</code> 634, 1904, 2117, 2920

L

<code>\label</code> 976, 985, 993, 1912
<code>\leavevmode</code> 2923
<code>\lst@caption</code> 2055
<code>\lst@label</code> 2052
<code>\lst@MakeCaption</code> 2051
<code>\LT@c@ption</code> 2043
<code>\ltx@backslashchar</code>
 744, 1548, 1607, 2135
<code>\ltx@car</code> 1653, 2226
<code>\ltx@cdr</code> 1655, 2229
<code>\ltx@empty</code> 291, 500, 566, 633, 720, 797, 1132, 1641, 1857, 1883, 1890, 1971, 1975,

M

<code>\m@ne</code> 1090
<code>\M@sect</code> 1991
<code>\M@TitleReference</code> 2371
<code>\mag</code> 1292, 2768, 2782, 2797, 2811
<code>\mainmatter</code> 60, 134
<code>\makeatletter</code> 11, 74, 101, 2153
<code>\makeatother</code> 16, 99
<code>\makebox</code> 169, 170
<code>\MakeRobustcommand</code> 233
<code>\marginparsep</code> 1333
<code>\marginparwidth</code> 1332
<code>\meaning</code> 2073
<code>\mediaheight</code> 1298
<code>\mediawidth</code> 1297
<code>\MessageBreak</code> 271, 514, 565, 571, 681, 1376, 1388, 1392, 1440, 1550, 1575, 1578, 1609, 1611, 1634, 1637, 1664, 1666, 1705, 1706, 2135, 2136, 2185, 2202, 2207, 2211, 2216, 2272, 2391, 2400, 2630, 2642, 2643

N		
<code>\NeedsTeXFormat</code>		<code>\protected@write</code> 716
. 3, 188, 220, 965, 1015, 1038,		<code>\providecommand</code> 223,
1065, 1119, 1159, 1250, 1268,		282, 1716, 2101, 2917, 2918, 2919
1449, 1620, 1713, 1731, 1844,		<code>\ProvidesPackage</code>
2088, 2491, 2503, 2554, 2897, 2952		... 189, 221, 966, 1016, 1039,
<code>\newcommand</code> 18, 78, 85, 91, 167, 975,		1066, 1120, 1160, 1251, 1269,
984, 992, 998, 1173, 1190, 1191,		1450, 1621, 1714, 1732, 1845,
1260, 1628, 2116, 2545, 2548,		2089, 2492, 2504, 2555, 2898, 2953
2570, 2577, 2584, 2588, 2637, 2920		
<code>\newcount</code> 2119		R
<code>\newcounter</code> 6, 911, 1027, 1743		<code>\read</code> 2235
<code>\newdimen</code> 2663, 2664		<code>\ReadonlyShipoutCounter</code> 1034
<code>\newif</code> 248, 701, 1856, 1869, 2104		<code>\refstepcounter</code> 1056
<code>\newlabel</code> 2308, 2319, 2350		<code>\renewcommand</code> 7, 46, 912
<code>\newmarks</code> 1637		<code>\RequirePackage</code> ... 191, 196, 224,
<code>\newpage</code> 143		225, 226, 227, 228, 231, 267,
<code>\nfss@text</code> 933		272, 280, 968, 1018, 1041, 1068,
<code>\NR@temp</code> 1974, 1975		1069, 1122, 1127, 1128, 1162,
<code>\number</code> 94, 109, 913, 925, 959,		1167, 1168, 1169, 1170, 1253,
1141, 1285, 1286, 1301, 1302,		1258, 1259, 1271, 1276, 1277,
1305, 1306, 1309, 1310, 1313,		1452, 1457, 1465, 1466, 1481,
1314, 1718, 1790, 1816, 1823,		1493, 1502, 1503, 1623, 1734,
2571, 2573, 2578, 2580, 2585,		1739, 1847, 1852, 1853, 2091,
2589, 2639, 2651, 2818, 2869, 2926		2096, 2097, 2494, 2506, 2557,
<code>\numexpr</code> 94,		2562, 2567, 2900, 2905, 2907, 2955
109, 115, 927, 1143, 1204, 1718,		<code>\reset@font</code> 933
1790, 2265, 2276, 2310, 2395,		<code>\rightarrow</code> 45
2404, 2586, 2590, 2782, 2811, 2933		<code>\romannumeral</code> 583,
		805, 826, 1642, 2131, 2140, 2141
O		
<code>\oddsidemargin</code> 1325		S
<code>\on@line</code> 1720, 2200		<code>\section</code> 63, 137, 145
<code>\openin</code> 2182		<code>\setcounter</code> 919, 1030
P		<code>\setlength</code> 2929
<code>\PackageError</code>		<code>\SetupKeyvalOptions</code> 2105
... 259, 270, 295, 305, 490, 1632		<code>\ShipoutBoxDepth</code> 1346
<code>\PackageInfo</code> 292, 521,		<code>\ShipoutBoxHeight</code> 1345
1721, 2189, 2201, 2292, 2334, 2355		<code>\ShipoutBoxWidth</code> 1344
<code>\PackageWarning</code> 346, 364,		<code>\space</code> 779, 1476,
381, 399, 616, 680, 1662, 1703, 2184		1488, 2203, 2204, 2208, 2212, 2217
<code>\PackageWarningNoLine</code> 2271, 2390, 2399		<code>\spinemargin</code> 1337
<code>\page</code> 2432		<code>\stepcounter</code>
<code>\pageheight</code> 1309, 1310, 2708		. 19, 1032, 1745, 1746, 1755, 1775
<code>\pagestyle</code> 49		<code>\stockheight</code> 1296, 2664
<code>\pagewidth</code> 1313, 1314		<code>\stockwidth</code>
<code>\paperheight</code> 1294		1295, 2663, 2668, 2686, 2696, 2709
<code>\paperwidth</code> 1293		T
<code>\pdf@escapehex</code> 1511		<code>\tableofcontents</code> 59, 132
<code>\pdf@strcmp</code> 477		<code>\tex</code> 2513, 2514, 2518
<code>\pdf@unescapehex</code> 1512		<code>\textheight</code> 1328
<code>\pdfhorigin</code> 1319		<code>\textwidth</code> 1327
<code>\pdfpageattr</code> 1476, 1482		<code>\the</code> 13, 153, 155, 430, 445, 461, 557,
<code>\pdfpageheight</code> 1318, 2695		562, 621, 627, 743, 757, 927,
<code>\pdfpagesattr</code> 1488, 1494		1034, 1083, 1096, 1137, 1143,
<code>\pdfpagewidth</code> 1317		1204, 1360, 1380, 1387, 1388,
<code>\pdfvariable</code>		1389, 1391, 1436, 1438, 1439,
1301, 1302, 1305, 1306, 1467, 1469		1467, 1469, 1482, 1494, 1520,
<code>\pdfvorigin</code> 1320		1555, 1573, 1574, 1576, 1866,
<code>\ProcessOptions</code> 217		2259, 2265, 2276, 2310, 2395, 2404
<code>\protect</code> 777		<code>\thechapter</code> 14
<code>\protected</code> 239		<code>\thefoo</code> 7, 12, 20
		<code>\thetype</code> 2322

<code>\thepage</code>	43, 44, 45, 714, 718, 779, 941, 1814	<code>\zlabel</code>	11, 83, 104, 138, 146, 973
<code>\thezpage</code>	16, 1814, 1820	<code>\zlistpageattr</code>	1515
<code>\thezref@unique</code>	10, <u>912</u> , 1811, 1812, 1819, 1820, 1822, 2928, 2930, 2933, 2934, 2942	<code>\zlistpagelayout</code>	15, 1355
<code>\title</code>	2441, 2470	<code>\zmakeperpage</code>	16, <u>1782</u>
<code>\toks@</code>	424, 430, 444, 445, 554, 557, 559, 562, 613, 620, 621, 627, 741, 743, 756, 757, 1079, 1083, 1091, 1096, 1376, 1380, 1386, 1387, 1548, 1555, 1572, 1573, 1860, 1866, 2244, 2259	<code>\znexpage</code>	13, 51, 54, 1187
<code>\topmargin</code>	1324	<code>\znexpagesetup</code>	14, 42, 1173
<code>\TR@TitleReference</code> ..	2366, 2426, 2465	<code>\znonexpagename</code>	46, 1190, 1238
<code>\trimedge</code>	1336	<code>\zpageref</code>	11, 126, <u>1007</u>
<code>\trintop</code>	1339	<code>\zposx</code>	18, 153, <u>2545</u> , 2933, 2934
<code>\ttl@sect@i</code>	2024	<code>\zposy</code>	18, 155, <u>2545</u>
<code>\ttl@straight@i</code>	2030	<code>\zref</code>	11, 25, 26, 27, 28, 112, 114, 123, 128, 129, 139, <u>998</u> , 1008
U		<code>\ZREF@@@delprop</code> ..	435, <u>437</u> , 472, 474
<code>\unexpanded</code>	271, 276	<code>\ZREF@@@newprop</code>	539, 543
<code>\UniqueCounterCall</code>	1188	<code>\ZREF@@delprop</code>	425, <u>434</u> , 448, 465, <u>471</u> , 481
<code>\UniqueCounterNew</code>	1171	<code>\ZREF@@extract</code>	809, <u>815</u>
<code>\uppermargin</code>	1340	<code>\ZREF@@makeperpage</code> ..	1784, 1790, 1794
<code>\url</code>	2461	<code>\ZREF@@newprop</code>	517, 528, 531, <u>535</u>
<code>\urluse</code>	2456	<code>\ZREF@@perpage@step</code>	1799, <u>1807</u>
<code>\usepackage</code>	9, 41, 48, 70, 72	<code>\ZREF@abspos@body@x@center</code>	<u>2848</u>
V		<code>\ZREF@abspos@body@x@left</code> ..	<u>2846</u> , <u>2873</u>
<code>\value</code>	13, 1137, 1360, 1520	<code>\ZREF@abspos@body@x@right</code> ..	<u>2847</u> , <u>2870</u>
<code>\verb</code>	173	<code>\ZREF@abspos@body@y@bottom</code>	<u>2853</u> , <u>2865</u> , <u>2887</u>
<code>\voffset</code>	1323	<code>\ZREF@abspos@body@y@center</code> ..	<u>2857</u> , <u>2888</u>
W		<code>\ZREF@abspos@body@y@top</code>	<u>2849</u> , <u>2854</u> , <u>2858</u> , <u>2886</u>
<code>\write</code>	705, 706, 1723	<code>\ZREF@abspos@foot@x@center</code>	<u>2863</u>
X		<code>\ZREF@abspos@foot@x@left</code>	<u>2861</u>
<code>\x</code>	331, 336, 724, 729, 898, 900, 1230, 1246, 1282, 1289, 1379, 1382, 1385, 1425, 1553, 1558, 1571, 1581, 1686, 1691, 1694, 1700, 1789, 1792, 2070, 2071, 2076, 2243, 2246, 2251, 2256, 2266, 2269, 2279, 2280, 2311, 2316	<code>\ZREF@abspos@foot@x@right</code>	<u>2862</u>
<code>\XR@ext</code>	2102	<code>\ZREF@abspos@foot@y@bottom</code>	<u>2864</u>
Y		<code>\ZREF@abspos@head@x@center</code>	<u>2828</u> , <u>2848</u> , <u>2863</u>
<code>\y</code>	2072, 2073	<code>\ZREF@abspos@head@x@left</code>	<u>2814</u> , <u>2825</u> , <u>2829</u> , <u>2846</u> , <u>2861</u>
Z		<code>\ZREF@abspos@head@x@right</code>	<u>2824</u> , <u>2847</u> , <u>2862</u>
<code>\z</code>	2073, 2074, 2076	<code>\ZREF@abspos@head@y@bottom</code> ..	<u>2838</u> , <u>2850</u>
<code>\z@</code>	2668, 2686, 2696, 2709, 2944	<code>\ZREF@abspos@head@y@center</code>	<u>2842</u>
<code>\Z@D@page</code>	1191	<code>\ZREF@abspos@head@y@top</code>	<u>2832</u> , <u>2839</u> , <u>2843</u>
<code>\Z@L@LastPage</code>	1081, 1093	<code>\ZREF@abspos@marginpar@x@center</code> ..	<u>2882</u>
<code>\Z@L@main</code>	1080, 1092	<code>\ZREF@abspos@marginpar@x@left</code> ..	<u>2868</u> , <u>2879</u> , <u>2883</u>
<code>\Z@L@ZREF@temp</code>	633, 637, 640, 651, 661	<code>\ZREF@abspos@marginpar@x@right</code> ..	<u>2878</u>
<code>\zdotfill</code>	19, 170, 173, <u>2921</u>	<code>\ZREF@abspos@marginpar@y@bottom</code> ..	<u>2887</u>
<code>\zdotfillsetup</code>	19, <u>2920</u>	<code>\ZREF@abspos@marginpar@y@center</code> ..	<u>2888</u>
<code>\zexternaldocument</code>	19, <u>2147</u>	<code>\ZREF@abspos@marginpar@y@top</code> ..	<u>2886</u>
<code>\ziflastpage</code>	12, <u>1109</u>	<code>\ZREF@abspos@media@height</code>	<u>2683</u> , <u>2729</u> , <u>2735</u>
<code>\zifrefundefined</code>	8, <u>763</u>	<code>\ZREF@abspos@media@width</code>	<u>2665</u> , <u>2722</u> , <u>2726</u>
<code>\zkvlabel</code>	<u>992</u>	<code>\ZREF@abspos@media@x@center</code>	<u>2724</u>
		<code>\ZREF@abspos@media@x@left</code>	<u>2718</u> , <u>2725</u>
		<code>\ZREF@abspos@media@x@right</code>	<u>2721</u>
		<code>\ZREF@abspos@media@y@bottom</code>	<u>2731</u>
		<code>\ZREF@abspos@media@y@center</code>	<u>2734</u>
		<code>\ZREF@abspos@media@y@top</code> ..	<u>2728</u> , <u>2746</u>
		<code>\ZREF@abspos@origin@x</code> ..	<u>2755</u> , <u>2816</u>
		<code>\ZREF@abspos@origin@y</code> ..	<u>2785</u> , <u>2834</u>

<code>\ZREF@abspos@paper@x@center</code>	<code>\zref@extract</code>
. 2743 , 2891	7 , 95 , 96 , 109 ,
<code>\ZREF@abspos@paper@x@left</code>	140 , 804 , 824 , 853 , 871 , 876 ,
. 2737 , 2815 , 2889	1005 , 1146 , 1242 , 1391 , 1438 ,
<code>\ZREF@abspos@paper@x@right</code> 2740 , 2890	1439 , 1562 , 1599 , 1819 , 1820 ,
<code>\ZREF@abspos@paper@y@bottom</code>	1913 , 2546 , 2549 , 2722 , 2726 ,
. 2747 , 2893	2729 , 2735 , 2741 , 2744 , 2749 ,
<code>\ZREF@abspos@paper@y@center</code>	2753 , 2774 , 2803 , 2817 , 2819 ,
. 2751 , 2894	2821 , 2826 , 2830 , 2835 , 2836 ,
<code>\ZREF@abspos@paper@y@top</code>	2840 , 2844 , 2851 , 2855 , 2859 ,
. 2746 , 2748 , 2752 , 2833 , 2892	2866 , 2871 , 2874 , 2875 , 2880 , 2884
<code>\ZREF@abspos@stock@x@center</code> 2891	<code>\ZREF@extractdefault</code> 825 , 841 , 844 , 875
<code>\ZREF@abspos@stock@x@left</code> 2889	<code>\zref@extractdefault</code> 7 , 116 , 117 ,
<code>\ZREF@abspos@stock@x@right</code> 2890	817 , 844 , 864 , 870 , 875 , 1102 ,
<code>\ZREF@abspos@stock@y@bottom</code> 2893	1103 , 1200 , 1215 , 1261 , 1822 ,
<code>\ZREF@abspos@stock@y@center</code> 2894	2364 , 2367 , 2368 , 2372 , 2373 ,
<code>\ZREF@abspos@stock@y@top</code> 2892	2376 , 2378 , 2379 , 2381 , 2383 ,
<code>\ZREF@abspos@used</code> 2615 , 2617	2573 , 2580 , 2623 , 2651 , 2818 , 2869
<code>\ZREF@absposnum</code> 2586 , 2590 , 2592	<code>\ZREF@false</code> 677 , 687
<code>\zref@absposnumused</code> 2637	<code>\ZREF@foundfalse</code> 2425
<code>\zref@absposnumx</code> 2572 , 2584 , 2606	<code>\ZREF@foundtrue</code> 2472
<code>\zref@absposnumy</code> 2579 , 2588 , 2609	<code>\ZREF@getcurrent</code> 582 , 593 , 596 , 874
<code>\zref@absposused</code> 2614	<code>\zref@getcurrent</code> 7 , 596 , 869 , 874
<code>\zref@absposx</code> 2570 , 2600	<code>\zref@hex</code>
<code>\zref@absposy</code> 2577 , 2603	1467 , 1469 , 1482 , 1494 , 1508 , 1511
<code>\zref@addprop</code>	<code>\zref@ifabsposnumundefined</code> 2650 , 2655
. 6 , 76 , 360 , 1035 , 1049 , 1053 ,	<code>\zref@ifabsposundefined</code> 2648
1130 , 1133 , 1287 , 1303 , 1307 ,	<code>\ZREF@ifDefinable</code> 242 , 763 ,
1311 , 1315 , 1468 , 1470 , 1483 ,	1007 , 1010 , 1109 , 1151 , 1187 ,
1495 , 1670 , 1855 , 2500 , 2568 , 2569	1355 , 1515 , 1782 , 1836 , 1903 ,
<code>\zref@addprops</code>	1906 , 2147 , 2530 , 2535 , 2540 , 2921
. 6 , 15 , 341 , 942 , 1353 , 1742 , 2515	<code>\ZREF@iflastpage</code> 1110 , 1112 , 1112
<code>\ZREF@addtoks</code> 755	<code>\zref@iflastpage</code> 12 , 1101 , 1115
<code>\ZREF@base@ok</code> 962 , 969 , 1019 ,	<code>\zref@iflistcontainsprop</code> 6 ,
1042 , 1070 , 1123 , 1163 , 1254 ,	310 , 345 , 363 , 380 , 398 , 648 , 658
1272 , 1453 , 1624 , 1735 , 1848 ,	<code>\zref@iflistundefined</code>
2092 , 2495 , 2507 , 2558 , 2901 , 2956 6 , 289 , 300 , 304 , 311
<code>\ZREF@call</code> 1196 , 1211 , 1220 , 1224 , 1232	<code>\zref@ifpropundefined</code> 7 , 485 , 489 ,
<code>\ZREF@def@abspos</code>	519 , 548 , 615 , 828 , 1434 , 1683 , 2419
. 2600 , 2603 , 2606 , 2609 , 2611	<code>\ZREF@ifrefcontainsprop</code> 787 , 795
<code>\zref@def@absposnumx</code> 2605	<code>\zref@ifrefcontainsprop</code>
<code>\ZREF@def@absposnumy</code> 2611 8 , 783 , 1436 , 2380 , 2476 , 2477
<code>\zref@def@absposnumy</code> 2608	<code>\ZREF@ifrefundefined</code>
<code>\zref@def@absposx</code> 2599 765 , 768 , 1197 , 1208 , 1218
<code>\zref@def@absposy</code> 2602	<code>\zref@ifrefundefined</code> 8 , 760 ,
<code>\ZREF@def@extract</code> 846 , 848	770 , 776 , 784 , 827 , 1209 , 1389 ,
<code>\zref@def@extract</code> 8 , 845	1576 , 1812 , 2619 , 2649 , 2657 , 2930
<code>\ZREF@def@extractdefault</code> 857 , 859	<code>\ZREF@immediatefalse</code> 678
<code>\zref@def@extractdefault</code> 856	<code>\ZREF@immediatetrue</code> 675 , 704
<code>\ZREF@default</code> 562 , 563 , 572	<code>\ZREF@label</code>
<code>\zref@default</code> 8 , 531 , 807 , 930 , 932	603 , 627 , 637 , 640 , 710 , 1083 , 1096
<code>\ZREF@delprop</code>	<code>\zref@label</code> 7 , 597 , 979 , 988
. 413 , 416 , 418 , 453 , 456 , 458	<code>\zref@labelbykv</code> 630 , 996
<code>\zref@delprop</code> 412 , 452	<code>\zref@labelbylist</code>
<code>\ZREF@df@dot</code> 2915 , 2919 , 2948 7 , 598 , 600 , 1137 , 1811 , 2532
<code>\ZREF@df@min</code> 2912 , 2917 , 2935	<code>\zref@labelbyprops</code> 7 , 88 ,
<code>\ZREF@df@unit</code> 2909 , 2918 , 2929	610 , 1195 , 2537 , 2542 , 2928 , 2942
<code>\ZREF@dotfill</code> 2931 , 2938 , 2947	<code>\zref@listexists</code> 6 , 303 , 322 ,
<code>\ZREF@ENV@line</code> 2966 , 2969	342 , 361 , 377 , 396 , 419 , 459 , 602
<code>\zref@env@line</code> 2961 , 2962	<code>\zref@listforloop</code> 321 , 657
<code>\ZREF@extract</code> 804 , 821 , 824 , 876	<code>\zref@listpageattr</code> 1515
	<code>\zref@listpagelayout</code> 1355
	<code>\zref@localaddprop</code> 395

<code>\zref@localaddprops</code>	376	<code>\ZREF@org@refstepcounter</code>	1058
<code>\zref@localdelprop</code>	415, 455, 669	<code>\ZREF@org@stepcounter</code> 1745, 1750, 1775	
<code>\ZREF@mainlist</code>	598, 936, 939, 942, 1035, 1049, 1053, 1855, 2500	<code>\ZREF@org@testdef</code>	
<code>\ZREF@makeperpage@opt</code> ...	1784, 1787	1362, 1364, 1522, 1524
<code>\ZREF@MARKS@DefineProp</code>		<code>\ZREF@org@thepage</code>	714, 718
.....	1646, 1647, 1648, 1682	<code>\ZREF@org@ttl@sect@i</code>	2026
<code>\zref@marks@register</code>		<code>\ZREF@org@ttl@straight@i</code>	2037
.....	1628, 1633, 1665, 1704	<code>\ZREF@org@write</code>	705, 706
<code>\ZREF@name</code>	229, 259, 270, 292, 295, 305, 346, 364, 381, 399, 490, 509, 521, 525, 549, 564, 616, 680, 692, 1632	<code>\ZREF@P</code>	505, 506, 508, 510, 519, 522, 526, 536, 537, 539, 540, 541, 545, 721, 725, 726, 735, 739, 744, 745
<code>\ZREF@NAME@bot</code>	1661, 1681	<code>\ZREF@pa@AfterLastShipout</code> 1518, 1616	
<code>\ZREF@NAME@first</code>	1660, 1680	<code>\ZREF@pa@AtVeryEnd</code> .. 1531, 1534, 1605	
<code>\ZREF@NAME@top</code>	1659, 1679	<code>\ZREF@pa@ListPage</code>	1552, 1570
<code>\zref@newlabel</code>		<code>\ZREF@pa@listtrue</code>	1516
.....	7, 282, 285, 750, 2263, 2349	<code>\ZREF@page@max</code> .	1360, 1426, 1520, 1582
<code>\zref@newlist</code>	6, 288, 939, 1074, 1129, 1656, 1741, 2511	<code>\zref@pageattr</code>	1560
<code>\ZREF@newprop</code>	498, 501, 504	<code>\zref@pageattr@used</code>	1567
<code>\zref@newprop</code>	6, 12, 13, 14, 75, 495, 940, 941, 944, 951, 955, 959, 1034, 1048, 1052, 1284, 1300, 1304, 1308, 1312, 1344, 1345, 1346, 1349, 1350, 1351, 1467, 1469, 1482, 1494, 1687, 1695, 2098, 2099, 2100, 2420, 2513, 2514, 2960, 2961	<code>\ZREF@pagenum@last</code>	1214, 1217
<code>\ZREF@NewPropAnchor</code> ..	943, 2149, 2499	<code>\ZREF@pagenum@this</code>	
<code>\ZREF@NewPropPageValue</code> 958, 1131, 1740		1199, 1204, 1207, 1217, 1223
<code>\ZREF@NewPropTheotype</code>	954, 2320	<code>\ZREF@par</code>	508, 533
<code>\ZREF@NewPropTitle</code> ...	949, 1854, 2150	<code>\ZREF@param</code>	
<code>\ZREF@nextpage</code>	1188, 1192	422, 423, 442, 460, 477, 646, 647, 648, 652, 673, 674, 677, 682
<code>\ZREF@nil</code> 545, 797, 836, 2236, 2242, 2248, 2253, 2263, 2279, 2308, 2316, 2407, 2414, 2423, 2426, 2465		<code>\ZREF@patch</code> .	249, 1055, 1917, 1923, 1929, 1935, 1941, 1947, 1953, 1959, 1990, 2003, 2009, 2015, 2023, 2029, 2042, 2050, 2062, 2077
<code>\ZREF@NOVALUE</code>	803	<code>\zref@pdfpageattr</code>	
<code>\ZREF@novalue</code>	796, 797, 803	1478, 1560, 1566, 1577
<code>\ZREF@np@call@next</code> ..	1182, 1186, 1241	<code>\zref@pdfpageattr@used</code>	1479
<code>\ZREF@np@call@nonext</code> 1179, 1185, 1237		<code>\zref@pdfpagesattr</code> ..	1490, 1597, 1610
<code>\ZREF@np@call@unknown</code> 1175, 1184, 1233		<code>\zref@pdfpagesattr@used</code> .	1491, 1602
<code>\ZREF@np@setup@i</code>	1174, 1177	<code>\ZREF@pl@AfterLastShipout</code> 1358, 1445	
<code>\ZREF@np@setup@ii</code>	1178, 1181	<code>\ZREF@pl@AtVeryEnd</code>	1371, 1374
<code>\ZREF@number</code> 924, 1562, 1568, 1630, 2656		<code>\ZREF@pl@ListEntry</code>	
<code>\ZREF@org@@begintheorem</code>	2080	1393, 1394, 1395, 1396, 1397, 1398, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1419, 1420, 1421, 1422, 1433
<code>\ZREF@org@@caption</code>	1920	<code>\ZREF@pl@ListPage</code>	1378, 1384
<code>\ZREF@org@@chapter</code>	1932, 1988	<code>\ZREF@pl@listtrue</code>	1356
<code>\ZREF@org@@opargbegintheorem</code> ..	2065	<code>\zref@pos@label@used</code>	2629
<code>\ZREF@org@@part</code>	1926	<code>\zref@pos@num@used</code>	2643
<code>\ZREF@org@@schapter</code>	1950	<code>\zref@prop</code> 324, 332, 333, 337, 658, 662	
<code>\ZREF@org@@sect</code>	1938	<code>\zref@propexists</code>	7, 344, 362, 379, 397, 488, 578, 647, 668, 999
<code>\ZREF@org@@spart</code>	1944	<code>\ZREF@refname@next</code>	
<code>\ZREF@org@@ssect</code>	1956	1202, 1209, 1218, 1242
<code>\ZREF@org@@stpelt</code> ..	1771, 1776, 1780	<code>\ZREF@refname@this</code>	
<code>\ZREF@org@beamer@section</code>	2006	1194, 1195, 1197, 1200
<code>\ZREF@org@beamer@subsection</code> ...	2012	<code>\ZREF@RefPrefix</code> .	284, 286, 1366, 1526
<code>\ZREF@org@beamer@subsubsection</code> 2018		<code>\ZREF@refused</code>	773, 775
<code>\ZREF@org@descriptionlabel</code>	1962	<code>\zref@refused</code>	
<code>\ZREF@org@lst@MakeCaption</code>	2058	8, 769, 772, 849, 860, 1004, 1011, 1113, 1114, 1149, 1264, 1568, 1603, 1911, 2618, 2626, 2639
<code>\ZREF@org@LT@c@ption</code>	2044		
<code>\ZREF@org@M@sect</code>	1997		

<code>\zref@require@unique</code>	<code>\zref@titleref@stripperperiodtrue</code> 1870
..... 10, 909, 1744, 2906	<code>\ZREF@true</code>
<code>\ZREF@Robust</code>	674, 688
232,	<code>\ZREF@u@getcurrent</code>
238, 244, 285, 288, 303, 310,	591
341, 360, 376, 395, 412, 415,	<code>\zref@unhex</code>
452, 455, 488, 495, 547, 577,	1509, 1512, 1561, 1598
597, 600, 610, 630, 702, 772,	<code>\ZREF@UpdatePdfTeX</code>
845, 856, 867, 883, 909, 929,	247
935, 1148, 1567, 1602, 1858,	<code>\ZREF@value</code>
1871, 2599, 2602, 2605, 2608, 2614	557, 558, 571
<code>\ZREF@SavedEscapechar</code>	<code>\ZREF@wrapper@babel</code>
461, 468	900, 906
<code>\zref@savepos</code> ...	<code>\zref@wrapper@babel</code>
18, 2516, 2524, 2526	.. 10, 140, 765, 773, 846, 857,
<code>\ZREF@savepos@ok</code>	883, 979, 988, 996, 1000, 1110,
2551, 2563	1907, 2600, 2603, 2606, 2609, 2615
<code>\zref@setcurrent</code>	<code>\zref@wrapper@immediate</code>
..... 6, 81, 541, 577, 697, 1057 10, 87, 636, 702, 1095
<code>\zref@setdefault</code>	<code>\ZREF@wrapper@unexpanded</code> ...
8, 929, 932	867, 881
<code>\zref@setmainlist</code>	<code>\zref@wrapper@unexpanded</code>
9, 935 10, 868, 873, 878, 2359
<code>\zref@showprop</code>	<code>\ZREF@wu@extract</code>
547	819, 871
<code>\ZREF@STAR</code>	<code>\ZREF@wu@extractdefault</code> ...
1654, 1678	839, 870
<code>\ZREF@stripperperiod</code>	<code>\ZREF@wu@getcurrent</code>
1882, 1890	591, 869
<code>\ZREF@temp</code>	<code>\ZREF@X</code>
193, 200, 201, 202,	497, 500, 537
203, 204, 205, 206, 207, 208,	<code>\zref@xr@</code>
209, 210, 211, 212, 213, 214,	2113
215, 216, 233, 234, 440, 441,	<code>\ZREF@xr@AddUrl</code>
442, 720, 741, 742, 750, 1278,	2125, 2128
1292, 1293, 1294, 1295, 1296,	<code>\ZREF@xr@input</code>
1297, 1298, 1317, 1318, 1319,	2256, 2351
1320, 1322, 1323, 1324, 1325,	<code>\ZREF@xr@AddURL</code>
1326, 1327, 1328, 1329, 1330,	2121, 2174, 2453
1331, 1332, 1333, 1334, 1335,	<code>\ZREF@xr@checkfile</code> ..
1336, 1337, 1338, 1339, 1340,	2178, 2181, 2231
1341, 1365, 1366, 1458, 1471,	<code>\ZREF@xr@checkkey</code>
1484, 1496, 1499, 1504, 1505,	2409, 2418
1506, 1507, 1525, 1526, 1534,	<code>\ZREF@xr@checklist</code>
1535, 1536, 1537, 1590, 1591,	2279, 2407
1592, 1593, 1653, 1654, 2031,	<code>\zref@xr@ext</code>
2032, 2123, 2133, 2136, 2140,	20, 2101, 2169
2622, 2625, 2626, 2755, 2758,	<code>\ZREF@xr@externaldocument</code>
2763, 2767, 2769, 2772, 2785,
2788, 2793, 2796, 2798, 2801, 2926 2156, 2159, 2163
<code>\ZREF@TempName</code> ..	<code>\ZREF@xr@externalfile</code>
1629, 1641, 1642, 2166, 2167, 2285, 2327
1644, 1670, 1683, 1687, 1695, 1706	<code>\ZREF@xr@file</code> ...
<code>\ZREF@TempNum</code>	2167, 2182, 2185,
.....	2191, 2202, 2225, 2273, 2392, 2401
1630, 1631, 1635, 1642, 1684, 1697	<code>\ZREF@xr@filelist</code>
<code>\zref@thepage</code>	2165,
13, 1145, 1154	2223, 2226, 2228, 2229, 2257, 2258
<code>\zref@thepage@atbegshi@hook</code>	<code>\ZREF@xr@found</code> .
.....	2193, 2203, 2265, 2310
1132, 1136	<code>\ZREF@xr@graburl</code>
<code>\zref@thepage@name</code>	2169, 2171
.....	<code>\ZREF@xr@ignored@empty</code>
13, 1140, 1146, 1149, 1203 2194, 2206, 2208, 2275, 2276
<code>\zref@thepage@refused</code> ...	<code>\ZREF@xr@ignored@ltx</code>
1148, 1153 2196, 2215, 2217, 2403, 2404
<code>\ZREF@titleref</code>	<code>\ZREF@xr@ignored@zref</code>
1907, 1909 2195, 2210, 2212, 2394, 2395
<code>\zref@titleref@cleanup</code> ..	<code>\ZREF@xr@line</code> ..
1858, 1898	2235, 2236, 2248, 2253
<code>\zref@titleref@current</code>	<code>\ZREF@xr@list</code>
.....	2269, 2270
949, 1877, 1881, 1882, 1901	<code>\ZREF@xr@ltx@ignorewarning</code>
<code>\ZREF@titleref@hook</code>	2398
.....	<code>\ZREF@xr@newlabel</code>
1857, 1861, 1865, 1888	2251, 2350
<code>\zref@titleref@setcurrent</code>	<code>\ZREF@xr@prefix</code>
.....	2164, 2264,
1871, 1919, 1925, 1931, 1937,	2300, 2304, 2309, 2335, 2342, 2346
1943, 1949, 1955, 1961, 1969,	<code>\ZREF@xr@process@label</code> ..
1972, 1976, 1980, 1982, 1993,	2253, 2308
1995, 2005, 2011, 2017, 2025,	<code>\ZREF@xr@process@zreflabel</code> 2248, 2263
2033, 2035, 2045, 2054, 2064, 2079	<code>\ZREF@xr@processfile</code> ...
	2181, 2234
	<code>\ZREF@xr@processline</code>
	2236, 2242
	<code>\ZREF@xr@refname</code>
	2264, 2290, 2297, 2309, 2332, 2339
	<code>\ZREF@xr@relax</code>
	2352, 2439
	<code>\ZREF@xr@scanparams</code>
	2314, 2423
	<code>\ZREF@xr@scantitleref</code> ...
	2426, 2465
	<code>\ZREF@xr@temp</code>
	2438, 2439
	<code>\ZREF@xr@tempname</code> 2267, 2268, 2288,
	2293, 2304, 2312, 2313, 2330, 2346

<code>\ZREF@xr@temprefname</code>	<code>\ZREF@zref</code>	1000, 1003
..... 2268, 2280, 2282,	<code>\ZREF@zsavepos</code> .	<u>2521</u> , 2532, 2537, 2542
2298, 2313, 2315, 2322, 2325, 2340	<code>\zrefused</code>	11, 92, 93, 161, 162, 163, <u>1010</u>
<code>\ZREF@xr@theURL</code>	<code>\zruns</code>	16, <u>1716</u>
2131, 2133, 2135, 2141, 2176, 2457	<code>\zsavepos</code>	17, 157, 158, <u>2530</u>
<code>\ZREF@xr@tolabel</code> ...	<code>\zsaveposx</code>	17, <u>2535</u>
2304, 2346, <u>2353</u>	<code>\zsaveposy</code>	<u>2540</u>
<code>\ZREF@xr@URL</code> ...	<code>\zthepage</code>	13, <u>1151</u>
2119, 2129, 2130, 2131	<code>\ztitleref</code>	17, <u>1906</u>
<code>\ZREF@xr@url</code> ...	<code>\ztitlerefsetup</code>	17, <u>1891</u>
2172, 2174, 2175, 2483	<code>\ztotpages</code>	15, 124, <u>1260</u>
<code>\ZREF@xr@urlcheck</code> ..	<code>\zunknownnextpagename</code> .	14, 1191, 1234
2288, 2330, <u>2475</u>	<code>\zunmakeperpage</code>	16, <u>1836</u>
<code>\ZREF@xr@zref@ignorewarning</code>	<code>\zxrsetup</code>	19, <u>2116</u>
..... 2300, 2342, <u>2389</u>		
<code>\ZREF@xr@zref@newlabel</code> ..		
2246, 2349		
<code>\ZREF@xr@zref@labelfalse</code>		
2155		
<code>\ZREF@xr@zref@labeltrue</code>		
2158		