

The zref package

Heiko Oberdiek*

2023-09-14 v2.35

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>	41
6.5	Module <code>counter</code>	42
6.6	Module <code>lastpage</code>	42
6.7	Module <code>thepage</code>	43
6.8	Module <code>nextpage</code>	44
6.9	Module <code>totpages</code>	46
6.10	Module <code>pagelayout</code>	46
6.10.1	Define layout properties	46
6.11	Module <code>pageattr</code>	49
6.12	Module <code>marks</code>	53
6.13	Module <code>runs</code>	54
6.14	Module <code>perpage</code>	54
6.15	Module <code>titleref</code>	57
6.15.1	Implementation	57
6.15.2	User interface	58
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>	61
6.15.9	Package <code>listings</code>	61
6.15.10	Theorems	62
6.16	Module <code>xr</code>	62
6.17	Module <code>hyperref</code>	70
6.18	Module <code>savepos</code>	70
6.18.1	Identification	70
6.18.2	Availability	70
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	75
6.19.3	Paper	76
6.19.4	Origin	77
6.19.5	Header	78
6.19.6	Body	78
6.19.7	Footer	79
6.19.8	Marginal notes	79
6.19.9	Stock paper	80
6.20	Module <code>dotfill</code>	80
6.21	Module <code>env</code>	81
7	Installation	82
7.1	Download	82
7.2	Bundle installation	82
7.3	Package installation	82
7.4	Refresh file name databases	83
7.5	Some details for the interested	83
8	References	84
9	History	84
	[2006/02/20 v1.0]	84
	[2006/05/03 v1.1]	84
	[2006/05/25 v1.2]	84
	[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]	85
	[2007/04/22 v1.8]	85
	[2007/05/02 v1.9]	85
	[2007/05/06 v2.0]	85
	[2007/05/28 v2.1]	85
	[2008/09/21 v2.2]	85
	[2008/10/01 v2.3]	85
	[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]	86
	[2010/03/29 v2.9]	86
	[2010/04/08 v2.10]	86
	[2010/04/15 v2.11]	86
	[2010/04/17 v2.12]	86
	[2010/04/19 v2.13]	86
	[2010/04/22 v2.14]	86
	[2010/04/23 v2.15]	87
	[2010/04/28 v2.16]	87
	[2010/05/01 v2.17]	87
	[2010/05/13 v2.18]	87
	[2010/10/22 v2.19]	87
	[2011/02/12 v2.20]	87
	[2011/03/18 v2.21]	87
	[2011/10/05 v2.22]	88
	[2011/12/05 v2.23]	88

[2012/04/04 v2.24]	88
[2016/05/16 v2.25]	88
[2016/05/21 v2.26]	88
[2018/11/21 v2.27]	88
[2019/11/29 v2.28]	88
[2020-03-03 v2.29]	88
[2020-03-04 v2.30]	88
[2020-05-28 v2.31]	89
[2020-07-03 v2.32]	89
[2022-03-08 v2.33]	89
[2022-04-07 v2.34]	89
[2023-09-14 v.2.35]	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}
```

$\langle refname \rangle$ is the name that denoted references (the name used in `\label` and `\ref`). $\langle propname \rangle$ 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 suport.
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 compability 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.

```
\zunknownnextpagename
\znonextpagename
```

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

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

```
\znexpagesetup {<unknown>} {<no next>} {<next>}
```

According to the case (see `\znexpage`) macro `\znexpage` 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 `\znexpagesetup`. It expects the definition texts for these three cases of a macro with one argument. The default is

```
\znexpagesetup{#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 \znexpagesetup
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]{\znexpage}
52 \fancypagestyle{plain}{%
53   \fancyhf{}%
54   \fancyhead[LE,RO]{\znexpage}%
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 `pdfTeX` provides (and `XYTeX`). `pdfTeX` is able to tell the current position on the page. The page position is not instantly known. First the page must be constructed by `TeX`'s asynchronous output routine. Thus the time where the position is known is the page shipout time. Thus a reference system where the information is recorded in the first run and made available for use in the second run comes in handy.

```
\zsavepos {⟨refname⟩}
```

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

```
\zsaveposx {⟨refname⟩}  
\zsaveposy {⟨refname⟩}
```

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

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

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

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

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

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

<code>\zref@savepos</code>

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

Thus `\zsavepos` is basically `\zref@savepos` followed by `\zref@labelbylist{<refname>}{savepos}`. If `\TeXeTstate` is detected and enabled, `\savepos` also adds `\zref@savepos` at the end to support `\beginR` where the whatits are processed in reverse order. The property list `savepos` contains the properties `posx` and `posy`.

3.14 Module `abspos`

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

```
\zref@absposx{<label>}{<value>}{<position>}
\zref@absposy{<label>}{<value>}{<position>}
```

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

`<label>` is a label set with `\zlabel` or `\zsavepos` that allows to retrieve the absolute page number.

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

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

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

3.15 Module `dotfill`

`\zdotfill`

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

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

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

`min=<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.

totxlabel: 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`, 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 pdf \TeX 's "savepos" feature. The position on the page is not known before the page is constructed and shipped out. Therefore the position is stored in references and are available for calculations in the next \LaTeX compile run.

```

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

```

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

```

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

```

Test for module `\dotfill`.

```

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

```

186 \langle /example \rangle

6 Implementation

6.1 Package zref

6.1.1 Identification

```
187  $\langle$ *package $\rangle$ 
188 \NeedsTeXFormat{LaTeX2e}
189 \ProvidesPackage{zref}
190 [2023-09-14 v2.35 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  $\langle$ /package $\rangle$ 
```

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:

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

\backslash ZREF@: Internal macros.

\backslash Z@L@*listname*: The properties of the list \langle *listname* \rangle .

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

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

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

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

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

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

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

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

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

6.2.2 Identification

```
219 \*base
220 \NeedsTeXFormat{LaTeX2e}
221 \ProvidesPackage{zref-base}%
222 [2023-09-14 v2.35 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{Z@R}
```

`\zref@newlabel` For the implementation of `\zref@newlabel` we call the same internal macro `\@newl@bel` that is used in `\newlabel`. Thus we have for free:

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

```

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@immediatetrue
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.

```
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   [2023-09-14 v2.35 Module user for zref (HO)]%
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 \newcommand*\zlabel{%
974   \ifx\label\ltx@gobble
975     \expandafter\ltx@gobble
976   \else
977     \expandafter\zref@wrapper@babel\expandafter\zref@label
978   \fi
979 }%
```

`\zkvlabel`

```

980 \newcommand*\zkvlabel}[1]{%
981   \ifx\label\ltx@gobble
982     \expandafter\ltx@gobblethree
983   \fi
984   \zref@wrapper@babel{\zref@labelbykv{#1}}%
985 }%
```

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

```

986 \newcommand*\zref}[2][default]{% robust because of optional argument
987   \zref@propexists{#1}{%
988     \zref@wrapper@babel\ZREF@zref{#2}{#1}%
989   }%
990 }%
991 \def\ZREF@zref#1{%
992   \zref@refused{#1}%
993   \zref@extract{#1}%
994 }%
```

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

```

995 \ZREF@IfDefinable\zpageref\def{%
996   {\zref[page]}%
997 }
```

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

```

998 \ZREF@IfDefinable\zrefused\def{%
999   {\zref@refused}%
1000 }

1001 </user>
```

6.4 Module `abspage`

```

1002 <*abspage>
1003 \NeedsTeXFormat{LaTeX2e}
1004 \ProvidesPackage{zref-abspage}%
1005   [2023-09-14 v2.35 Module abspage for zref (HO)]%
1006 \RequirePackage{zref-base}[2019/11/29]
1007 \ifx\ZREF@base@ok Y%
1008 \else
1009   \expandafter\endinput
1010 \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.

```

1011 \begingroup
1012 \@ifundefined{c@abspage}
1013   {%
1014   \let\@addtoreset\ltx@gobbletwo
1015   \newcounter{abspage}%
1016   }{}
1017 \endgroup
1018 \setcounter{abspage}{0}%
1019 \AddToHook{shipout/before}{%
1020   \stepcounter{abspage}%
1021 }%
1022 \zref@newprop*{abspage}[0]{\the\ReadonlyShipoutCounter}%
1023 \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.

```
1024 </abspage>
```

6.5 Module counter

```

1025 <*counter>
1026 \NeedsTeXFormat{LaTeX2e}
1027 \ProvidesPackage{zref-counter}%
1028   [2023-09-14 v2.35 Module counter for zref (HO)]%
1029 \RequirePackage{zref-base}[2019/11/29]
1030 \ifx\ZREF@base@ok Y%
1031 \else
1032   \expandafter\endinput
1033 \fi

```

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

```

1034 \@ifl@t@r\fmtversion{2020-10-01}
1035 {
1036   \zref@newprop{counter}{\@currentcounter}
1037   \zref@addprop\ZREF@mainlist{counter}
1038 }
1039 {
1040   \zref@newprop{counter}{}
1041   \zref@addprop\ZREF@mainlist{counter}
1042   \AtBeginDocument{%
1043     \ZREF@patch{refstepcounter}{%
1044       \def\refstepcounter#1{%
1045         \zref@setcurrent{counter}{#1}%
1046         \ZREF@org@refstepcounter{#1}%
1047       }%
1048     }%
1049   }
1050 }
1051 </counter>

```

6.6 Module lastpage

```

1052 <*lastpage>
1053 \NeedsTeXFormat{LaTeX2e}
1054 \ProvidesPackage{zref-lastpage}%
1055   [2023-09-14 v2.35 Module lastpage for zref (HO)]%
1056 \RequirePackage{zref-base}[2019/11/29]

```

```

1057 \RequirePackage{zref-abspage} [2019/11/29]
1058 \ifx\ZREF@base@ok Y%
1059 \else
1060 \expandafter\endinput
1061 \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.

```

1062 \zref@newlist{LastPage}
1063 \AddToHook {enddocument/afterlastpage}{%
1064 \if@filesw
1065 \begingroup
1066 \advance\c@page\m@ne
1067 \toks@\expandafter\expandafter\expandafter{%
1068 \expandafter\Z@L@main
1069 \Z@L@LastPage
1070 }%
1071 \expandafter\zref@wrapper@immediate\expandafter{%
1072 \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
1073 }%
1074 \endgroup
1075 \fi
1076 }

```

`\zref@iflastpage`

```

1077 \def\zref@iflastpage#1{%
1078 \ifnum\zref@extractdefault{#1}{abspage}{-1}=%
1079 \zref@extractdefault{LastPage}{abspage}{-2} %
1080 \expandafter\ltx@firstoftwo
1081 \else
1082 \expandafter\ltx@secondoftwo
1083 \fi
1084 }

```

`\ziflastpage`

```

1085 \ZREF@IfDefinable\ziflastpage\def{%
1086 {\zref@wrapper@babel\ZREF@iflastpage}%
1087 }

```

`ZREF@iflastpage`

```

1088 \def\ZREF@iflastpage#1{%
1089 \zref@refused{LastPage}%
1090 \zref@refused{#1}%
1091 \zref@iflastpage{#1}%
1092 }

```

1093 `\lastpage`)

6.7 Module `thepage`

```

1094 <{*thepage}
1095 \NeedsTeXFormat{LaTeX2e}
1096 \ProvidesPackage{zref-thepage}%
1097 [2023-09-14 v2.35 Module thepage for zref (HO)]%
1098 \RequirePackage{zref-base} [2019/11/29]
1099 \ifx\ZREF@base@ok Y%
1100 \else
1101 \expandafter\endinput
1102 \fi

1103 \RequirePackage{atbegshi} [2011/10/05]
1104 \RequirePackage{zref-abspage} [2019/11/29]

```

```

1105 \zref@newlist{thepage}
1106 \zref@addprop{thepage}{page}
1107 \ZREF@NewPropPageValue

\zref@thepage@atbegshi@hook
1108 \let\zref@thepage@atbegshi@hook\ltx@empty

1109 \zref@addprop{thepage}{pagevalue}
1110 \AddToHook{shipout/before}{%
1111   \AtBeginShipoutAddToBox{%
1112     \zref@thepage@atbegshi@hook
1113     \zref@labelbylist{thepage\the\value{abspage}}{thepage}%
1114   }%
1115 }

\zref@thepage@name
1116 \ltx@ifundefined{numexpr}{%
1117   \def\zref@thepage@name#1{thepage\number#1}%
1118 }{%
1119   \def\zref@thepage@name#1{thepage\the\numexpr#1}%
1120 }

\zref@thepage
1121 \def\zref@thepage#1{%
1122   \zref@extract{\zref@thepage@name{#1}}{page}%
1123 }%

\zref@thepage@refused
1124 \ZREF@Robust\def\zref@thepage@refused#1{%
1125   \zref@refused{\zref@thepage@name{#1}}%
1126 }%

\zthepage
1127 \ZREF@ifdefinable\zthepage\def{%
1128   #1{%
1129     \zref@thepage@refused{#1}%
1130     \zref@thepage{#1}%
1131   }%
1132 }

1133 </thepage>

```

6.8 Module nextpage

```

1134 <*nextpage>
1135 \NeedsTeXFormat{LaTeX2e}
1136 \ProvidesPackage{zref-nextpage}%
1137 [2023-09-14 v2.35 Module nextpage for zref (HO)]%
1138 \RequirePackage{zref-base}[2019/11/29]
1139 \ifx\ZREF@base@ok Y%
1140 \else
1141   \expandafter\endinput
1142 \fi

1143 \RequirePackage{zref-abspage}[2019/11/29]
1144 \RequirePackage{zref-thepage}[2019/11/29]
1145 \RequirePackage{zref-lastpage}[2019/11/29]
1146 \RequirePackage{uniquecounter}[2009/12/18]

1147 \UniqueCounterNew{znextpage}
1148
1149 \newcommand*{\znextpagesetup}{%
1150   \afterassignment\ZREF@np@setup@i
1151   \def\ZREF@np@call@unknown##1%

```

```

1152 }
1153 \def\ZREF@np@setup@i{%
1154   \afterassignment\ZREF@np@setup@ii
1155   \def\ZREF@np@call@nonext##1%
1156 }
1157 \def\ZREF@np@setup@ii{%
1158   \def\ZREF@np@call@next##1%
1159 }
1160 \def\ZREF@np@call@unknown#1{#1}
1161 \def\ZREF@np@call@nonext#1{#1}
1162 \def\ZREF@np@call@next#1{#1}
1163 \ZREF@ifDefinable\znextpage\def{%
1164   {\UniqueCounterCall{znextpage}{\ZREF@nextpage}}%
1165 }%
1166 \newcommand*\znonextpagename{}
1167 \newcommand*\zunknownnextpagename{\Z@D@page}
1168 \def\ZREF@nextpage#1{%
1169   \begingroup
1170     \def\ZREF@refname@this{zref@np#1}%
1171     \zref@labelbyprops\ZREF@refname@this{abspage}%
1172     \chardef\ZREF@call=0 % unknown
1173     \ZREF@ifrefundefined\ZREF@refname@this{%
1174       }{%
1175         \edef\ZREF@pagenum@this{%
1176           \zref@extractdefault\ZREF@refname@this{abspage}{0}%
1177         }%
1178         \edef\ZREF@refname@next{%
1179           \zref@thepage@name{%
1180             \the\numexpr\ZREF@pagenum@this+1%
1181           }%
1182         }%
1183         \ifnum\ZREF@pagenum@this>0 %
1184           \ZREF@ifrefundefined{LastPage}{%
1185             \zref@ifrefundefined\ZREF@refname@next{%
1186               }{%
1187                 \chardef\ZREF@call=2 % next page
1188               }%
1189             }{%
1190               \edef\ZREF@pagenum@last{%
1191                 \zref@extractdefault{LastPage}{abspage}{0}%
1192               }%
1193               \ifnum\ZREF@pagenum@this<\ZREF@pagenum@last\ltx@space
1194                 \ZREF@ifrefundefined\ZREF@refname@next{%
1195                   }{%
1196                     \chardef\ZREF@call=2 % next page
1197                   }%
1198                 \else
1199                   \ifnum\ZREF@pagenum@this=\ZREF@pagenum@this\ltx@space
1200                     \chardef\ZREF@call=1 % no next page
1201                   \fi
1202                 \fi
1203               }%
1204             \fi
1205           }%
1206         \edef\x{%
1207           \endgroup
1208           \ifcase\ZREF@call
1209             \noexpand\ZREF@np@call@unknown{%
1210               \noexpand\zunknownnextpagename
1211             }%
1212           \or
1213             \noexpand\ZREF@np@call@nonext{%

```

```

1214         \noexpand\znonextpagename
1215     }%
1216     \else
1217         \noexpand\ZREF@np@call@next{%
1218         \noexpand\zref@extract{\ZREF@refname@next}{page}%
1219     }%
1220     \fi
1221 }%
1222 \x
1223 }
1224 </nextpage>

```

6.9 Module `totpages`

```

1225 <*totpages>
1226 \NeedsTeXFormat{LaTeX2e}
1227 \ProvidesPackage{zref-totpages}%
1228 [2023-09-14 v2.35 Module totpages for zref (HO)]%
1229 \RequirePackage{zref-base}[2019/11/29]
1230 \ifx\ZREF@base@ok Y%
1231 \else
1232 \expandafter\endinput
1233 \fi

```

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

```

1234 \RequirePackage{zref-abspage}[2019/11/29]
1235 \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.

```

1236 \newcommand*\ztotpages{%
1237 \zref@extractdefault{LastPage}{abspage}{0}%
1238 }

```

Also we mark the reference `LastPage` as used:

```

1239 \AtBeginDocument{%
1240 \zref@refused{LastPage}%
1241 }
1242 </totpages>

```

6.10 Module `pagelayout`

```

1243 <*pagelayout>
1244 \NeedsTeXFormat{LaTeX2e}
1245 \ProvidesPackage{zref-pagelayout}%
1246 [2023-09-14 v2.35 Module pagelayout for zref (HO)]%
1247 \RequirePackage{zref-base}[2019/11/29]
1248 \ifx\ZREF@base@ok Y%
1249 \else
1250 \expandafter\endinput
1251 \fi

1252 \RequirePackage{zref-thepage}[2019/11/29]
1253 \RequirePackage{iftex}[2019/11/07]%

```

6.10.1 Define layout properties

```

1254 \def\ZREF@temp#1{%
1255 \begingroup
1256 \escapechar=-1 %
1257 \ltx@ifundefined{\string#1}{\endgroup}{%
1258 \edef\x{%
1259 \endgroup
1260 \noexpand\zref@newprop*\string#1}%
1261 [\noexpand\number\noexpand#1]% hash-ok

```

```

1262             {\noexpand\number\noexpand#1}%
1263     \noexpand\zref@addprop{thepage}{\string#1}%
1264     }%
1265     \x
1266     }%
1267 }
1268 \ZREF@temp\mag
1269 \ZREF@temp\paperwidth
1270 \ZREF@temp\paperheight
1271 \ZREF@temp\stockwidth % memoir.cls, crop.sty
1272 \ZREF@temp\stockheight % memoir.cls, crop.sty
1273 \ZREF@temp\mediawidth % VTeX
1274 \ZREF@temp\mediaheight % VTeX
1275 \ifluatex
1276 \zref@newprop*{pdfvorigin}%
1277             [\number\pdfvariable vorigin]% hash-ok
1278             {\number\pdfvariable vorigin}%
1279 \zref@addprop{thepage}{pdfvorigin}
1280 \zref@newprop*{pdfhorigin}%
1281             [\number\pdfvariable horigin]% hash-ok
1282             {\number\pdfvariable horigin}%
1283 \zref@addprop{thepage}{pdfhorigin}
1284 \zref@newprop*{pdfpageheight}%
1285             [\number\pageheight]% hash-ok
1286             {\number\pageheight}%
1287 \zref@addprop{thepage}{pdfpageheight}
1288 \zref@newprop*{pdfpagewidth}%
1289             [\number\pagewidth]% hash-ok
1290             {\number\pagewidth}%
1291 \zref@addprop{thepage}{pdfpagewidth}
1292 \else
1293 \ZREF@temp\pdfpagewidth
1294 \ZREF@temp\pdfpageheight
1295 \ZREF@temp\pdfhorigin
1296 \ZREF@temp\pdfvorigin
1297 \fi
1298 \ZREF@temp\hoffset
1299 \ZREF@temp\voffset
1300 \ZREF@temp\topmargin
1301 \ZREF@temp\oddsidemargin
1302 \ZREF@temp\evensidemargin
1303 \ZREF@temp\textwidth
1304 \ZREF@temp\textheight
1305 \ZREF@temp\headheight
1306 \ZREF@temp\headsep
1307 \ZREF@temp\footskip
1308 \ZREF@temp\marginparwidth
1309 \ZREF@temp\marginparsep
1310 \ZREF@temp\columnwidth
1311 \ZREF@temp\columnsep
1312 \ZREF@temp\trimedge % memoir.cls
1313 \ZREF@temp\spinemargin % memoir.cls
1314 \ZREF@temp\foremargin % memoir.cls
1315 \ZREF@temp\trimtop % memoir.cls
1316 \ZREF@temp\uppermargin % memoir.cls
1317 \ZREF@temp\headmargin % memoir.cls

1318 \IfFormatAtLeastTF{2020/10/01}
1319 {
1320     \zref@newprop*{outputboxwd}[Opt]{\ShipoutBoxWidth}
1321     \zref@newprop*{outputboxht}[Opt]{\ShipoutBoxHeight}
1322     \zref@newprop*{outputboxdp}[Opt]{\ShipoutBoxDepth}
1323 }

```

```

1324 {
1325   \zref@newprop*{outputboxwd}[Opt]{\AtBeginShipoutBoxWidth}
1326   \zref@newprop*{outputboxht}[Opt]{\AtBeginShipoutBoxHeight}
1327   \zref@newprop*{outputboxdp}[Opt]{\AtBeginShipoutBoxDepth}
1328 }
1329 \zref@addprops{thepage}{outputboxwd,outputboxht,outputboxdp}

\ifZREF@pl@list
1330 \ltx@newif\ifZREF@pl@list

\zref@listpagelayout
1331 \ZREF@IfDefinable\zlistpagelayout\def{%
1332   {\global\ZREF@pl@listtrue}%
1333 }

\ZREF@pl@AfterLastShipout
1334 \def\ZREF@pl@AfterLastShipout{%
1335   \ifZREF@pl@list
1336     \edef\ZREF@page@max{\the\value{abspage}}%
1337     \ltx@ifundefined{ZREF@org@testdef}{%
1338       \let\ZREF@org@testdef\@testdef
1339       \def\@testdef##1##2##3{%
1340         \ZREF@org@testdef{##1}{##2}{##3}%
1341         \def\ZREF@temp{##1}%
1342         \ifx\ZREF@temp\ZREF@RefPrefix
1343           \expandafter\gdef\cname##1@##2\endcname{##3}%
1344         \fi
1345       }%
1346     }{}%
1347     \AddToHook{enddocument/afteraux}{\ZREF@pl@AtVeryEnd}%
1348   \fi
1349 }

\ZREF@pl@AtVeryEnd
1350 \def\ZREF@pl@AtVeryEnd{%
1351   \begingroup
1352     \toks@{Page layout parameters:\MessageBreak}%
1353     \count@=1 %
1354     \ZREF@pl@ListPage
1355     \edef\x{\endgroup
1356       \noexpand\@PackageInfoNoLine{zref-pagelayout}{\the\toks@}%
1357     }%
1358   \x
1359 }

\ZREF@pl@ListPage
1360 \def\ZREF@pl@ListPage{%
1361   \edef\x{%
1362     \toks@={%
1363       \the\toks@
1364       Page \the\count@:\noexpand\MessageBreak
1365       \zref@ifrefundefined{thepage\the\count@}{-}{%
1366         \ltx@space\ltx@space mag = %
1367         \zref@extract{thepage\the\count@}{mag}%
1368         \noexpand\MessageBreak
1369         \ZREF@pl@ListEntry{paperwidth}%
1370         \ZREF@pl@ListEntry{paperheight}%
1371         \ZREF@pl@ListEntry{stockwidth}%
1372         \ZREF@pl@ListEntry{stockheight}%
1373         \ZREF@pl@ListEntry{mediawidth}%
1374         \ZREF@pl@ListEntry{mediaheight}%
1375         \ZREF@pl@ListEntry{pdfpagewidth}%

```



```

1376     \ZREF@pl@ListEntry{pdfpageheight}%
1377     \ZREF@pl@ListEntry{pdfhorigin}%
1378     \ZREF@pl@ListEntry{pdfvorigin}%
1379     \ZREF@pl@ListEntry{hoffset}%
1380     \ZREF@pl@ListEntry{voffset}%
1381     \ZREF@pl@ListEntry{topmargin}%
1382     \ZREF@pl@ListEntry{oddsidemargin}%
1383     \ZREF@pl@ListEntry{evensidemargin}%
1384     \ZREF@pl@ListEntry{textwidth}%
1385     \ZREF@pl@ListEntry{textheight}%
1386     \ZREF@pl@ListEntry{headheight}%
1387     \ZREF@pl@ListEntry{headsep}%
1388     \ZREF@pl@ListEntry{footskip}%
1389     \ZREF@pl@ListEntry{marginparwidth}%
1390     \ZREF@pl@ListEntry{marginparsep}%
1391     \ZREF@pl@ListEntry{columnwidth}%
1392     \ZREF@pl@ListEntry{columnsep}%
1393     \ZREF@pl@ListEntry{trimedge}%
1394     \ZREF@pl@ListEntry{spinemargin}%
1395     \ZREF@pl@ListEntry{foremargin}%
1396     \ZREF@pl@ListEntry{trimtop}%
1397     \ZREF@pl@ListEntry{uppermargin}%
1398     \ZREF@pl@ListEntry{headmargin}%
1399   }%
1400 }%
1401 }\x
1402 \ifnum\ZREF@page@max>\count@
1403   \advance\count@ by\ltx@one
1404 \else
1405   \expandafter\ltx@gobble
1406 \fi
1407 \ZREF@pl@ListPage
1408 }

```

\ZREF@pl@ListEntry

```

1409 \def\ZREF@pl@ListEntry#1{%
1410   \zref@ifpropundefined{#1}{%
1411     }{%
1412       \zref@ifrefcontainsprop{thepage\the\count@}{#1}{%
1413         \ltx@space\ltx@space#1 = %
1414         \zref@extract{thepage\the\count@}{#1}sp = %
1415         \the\dimexpr\zref@extract{thepage\the\count@}{#1}sp\relax
1416         \noexpand\MessageBreak
1417       }{%}
1418     }%
1419   }

1420 \AddToHook {enddocument/afterlastpage}{%
1421   \ZREF@pl@AfterLastShipout
1422 }

1423 </pagelayout>

```

6.11 Module pageattr

```

1424 <*pageattr>
1425 \NeedsTeXFormat{LaTeX2e}
1426 \ProvidesPackage{zref-pageattr}%
1427 [2023-09-14 v2.35 Module pageattr for zref (H0)]%
1428 \RequirePackage{zref-base}[2019/11/29]
1429 \ifx\ZREF@base@ok Y%
1430 \else
1431   \expandafter\endinput

```

```

1432 \fi
1433 \RequirePackage{iftex}[2019/11/07]%
1434 \let\ZREF@temp=N%
1435 \ifluatex
1436 \expandafter\@firstoftwo
1437 \else
1438 \expandafter\@secondoftwo
1439 \fi
1440 {%luatex
1441 \RequirePackage{zref-thepage}[2019/11/29]
1442 \RequirePackage{zref-lastpage}[2019/11/29]%
1443 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfvariable pageattr}}%
1444 \zref@addprop{thepage}{pdfpageattr}%
1445 \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfvariable pagesattr}}%
1446 \zref@addprop{LastPage}{pdfpagesattr}%
1447 \let\ZREF@temp=Y%
1448 }
1449 {%other
1450 \ltx@ifundefined{pdfpageattr}{%
1451 \PackageInfoNoLine{zref-pageattr}{%
1452 \string\pdfpageattr\space is not available%
1453 }%
1454 \def\zref@pdfpageattr#1{}%
1455 \def\zref@pdfpageattr@used#1{}%
1456 }{%
1457 \RequirePackage{zref-thepage}[2019/11/29]%
1458 \zref@newprop*{pdfpageattr}[]{\zref@hex{\the\pdfpageattr}}%
1459 \zref@addprop{thepage}{pdfpageattr}%
1460 \let\ZREF@temp=Y%
1461 }
1462 \ltx@ifundefined{pdfpagesattr}{%
1463 \PackageInfoNoLine{zref-pageattr}{%
1464 \string\pdfpagesattr\space is not available%
1465 }%
1466 \def\zref@pdfpagesattr{}%
1467 \def\zref@pdfpagesattr@used{}%
1468 }{%
1469 \RequirePackage{zref-lastpage}[2019/11/29]%
1470 \zref@newprop*{pdfpagesattr}[]{\zref@hex{\the\pdfpagesattr}}%
1471 \zref@addprop{LastPage}{pdfpagesattr}%
1472 \let\ZREF@temp=Y%
1473 }%
1474 }%

1475 \ifx\ZREF@temp N%
1476 \expandafter\endinput
1477 \fi

1478 \RequirePackage{zref-abspage}[2019/11/29]
1479 \RequirePackage{pdftexcmds}[2010/04/01]
1480 \let\ZREF@temp=Y%
1481 \ltx@ifundefined{pdf@escapehex}{\let\ZREF@temp=N}{%
1482 \ltx@ifundefined{pdf@unescapehex}{\let\ZREF@temp=N}{%
1483 \ifx\ZREF@temp N%
1484 \let\zref@hex\ltx@firstofone
1485 \let\zref@unhex\ltx@firstofone
1486 \else
1487 \let\zref@hex\pdf@escapehex
1488 \let\zref@unhex\pdf@unescapehex
1489 \fi

```

```
\ifZREF@pa@list
```

```
1490 \ltx@newif\ifZREF@pa@list
```

`\zref@listpageattr`

```
1491 \ZREF@ifDefinable\zlistpageattr\def{%
1492   {\ZREF@pa@listtrue}%
1493 }
```

`\ZREF@pa@AfterLastShipout`

```
1494 \def\ZREF@pa@AfterLastShipout{%
1495   \ifZREF@pa@list
1496     \edef\ZREF@page@max{\the\value{abspage}}%
1497     \ltx@ifundefined{ZREF@org@testdef}{%
1498       \let\ZREF@org@testdef\@testdef
1499       \def\@testdef##1##2##3{%
1500         \ZREF@org@testdef{##1}{##2}{##3}%
1501         \def\ZREF@temp{##1}%
1502         \ifx\ZREF@temp\ZREF@RefPrefix
1503           \expandafter\xdef\csname##1@##2\endcsname{##3}%
1504         \fi
1505       }%
1506     }{}%
1507     \AddToHook{enddocument/afteraux}{\ZREF@pa@AtVeryEnd}%
1508   \fi
1509 }
```

`\ZREF@pa@AtVeryEnd`

```
1510 \let\ZREF@temp=Y%
1511 \ltx@ifundefined{pdfpageattr}{\let\ZREF@temp=N}
1512 \ifluatex \let\ZREF@temp=N \fi
1513 \ifx\ZREF@temp Y
1514   \expandafter\@firstoftwo
1515 \else
1516   \expandafter\@secondoftwo
1517 \fi
1518 {%
1519   \def\ZREF@pa@AtVeryEnd{}%
1520 }
1521 {%
1522   \def\ZREF@pa@AtVeryEnd{%
1523     \begingroup
1524     \toks@{List of \ltx@backslashchar
1525       \ifluatex pdfvariable\else pdf\fi
1526       pdfpageattr:\MessageBreak}%
1527     \count@=1 %
1528     \ZREF@pa@ListPage
1529     \edef\x{\endgroup
1530       \noexpand\@PackageInfoNoLine{zref-pageattr}{%
1531         \the\toks@
1532       }%
1533     }%
1534     \x
1535   }%

```

`\zref@pageattr`

```
1536 \def\zref@pdfpageattr#1{%
1537   \zref@unhex{%
1538     \zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}%
1539   }%
1540 }
1541 % compability, \zref@pageattr was defined in older versions
1542 \let\zref@pageattr\zref@pdfpageattr
```

`\zref@pageattr@used`

```
1543 \ZREF@Robust\def\zref@pageattr@used#1{%
```

```

1544 \zref@refused{thepage\ZREF@number{#1}}%
1545 }

```

\ZREF@pa@ListPage

```

1546 \def\ZREF@pa@ListPage{%
1547   \edef\x{%
1548     \toks@={%
1549       \the\toks@
1550       Page \the\count@:%
1551       \noexpand\MessageBreak
1552       \zref@ifrefundefined{thepage\the\count@}{-}{%
1553         <<\zref@pdfpagesattr\count@>>%
1554         \noexpand\MessageBreak
1555       }%
1556     }%
1557   }\x
1558   \ifnum\ZREF@page@max>\count@
1559     \advance\count@ by\ltx@one
1560   \else
1561     \expandafter\ltx@gobble
1562   \fi
1563   \ZREF@pa@ListPage
1564 }%
1565 }

1566 \let\ZREF@temp=Y%
1567 \ltx@ifundefined{pdfpagesattr}{-}{\let\ZREF@temp=N}
1568 \ifluatex \let\ZREF@temp=N \fi
1569 \ifx\ZREF@temp N
1570   \expandafter\@firstofone
1571 \fi
1572 {%

```

\zref@pdfpagesattr

```

1573 \def\zref@pdfpagesattr{%
1574   \zref@unhex{%
1575     \zref@extract{LastPage}{pdfpagesattr}%
1576   }%
1577 }%

```

\zref@pdfpagesattr@used

```

1578 \ZREF@Robust\def\zref@pdfpagesattr@used{%
1579   \zref@refused{LastPage}%
1580 }%

1581 \ltx@LocalAppendToMacro\ZREF@pa@AtVeryEnd{%
1582   \@PackageInfoNoLine{zref-pageattr}{%
1583     \ltx@backslashchar
1584     \ifluatex pdfvariable\else pdf\fi
1585     pagesattr:\MessageBreak
1586     <<\zref@pdfpagesattr>>%
1587     \MessageBreak
1588   }%
1589 }%
1590 }
1591 \AddToHook {enddocument/afterlastpage}{%
1592   \ZREF@pa@AfterLastShipout
1593 }
1594 </pageattr>

```

6.12 Module marks

```
1595 <*marks>
1596 \NeedsTeXFormat{LaTeX2e}
1597 \ProvidesPackage{zref-marks}%
1598 [2023-09-14 v2.35 Module marks for zref (HO)]%
1599 \RequirePackage{zref-base}[2019/11/29]
1600 \ifx\ZREF@base@ok Y%
1601 \else
1602 \expandafter\endinput
1603 \fi

1604 \newcommand*{\zref@marks@register}[3][ ]{%
1605 \edef\ZREF@TempName{#1}%
1606 \edef\ZREF@TempNum{\ZREF@number{#2}}%
1607 \ifnum\ZREF@TempNum<\ltx@zero %
1608 \PackageError\ZREF@name{%
1609 \string\zref@marks@register\ltx@space is called with invalid%
1610 \MessageBreak
1611 marks register number (\ZREF@TempNum)%
1612 }{%
1613 Use ‘0’ or the command, defined by \string\newmarks.\MessageBreak
1614 \@ehc
1615 }%
1616 \else
1617 \ifx\ZREF@TempName\ltx@empty
1618 \edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}%
1619 \else
1620 \edef\ZREF@TempName{marks\ZREF@TempName}%
1621 \fi
1622 \ZREF@MARKS@DefineProp{top}%
1623 \ZREF@MARKS@DefineProp{first}%
1624 \ZREF@MARKS@DefineProp{bot}%
1625 \kv@parse{#3}{%
1626 \ifx\kv@value\relax
1627 \def\kv@value{top,first,bot}%
1628 \fi
1629 \edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}%
1630 \ifx\ZREF@temp\ZREF@STAR
1631 \edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}%
1632 \zref@newlist\kv@key
1633 \fi
1634 \expandafter\comma@parse\expandafter{\kv@value}{%
1635 \ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else
1636 \ifx\comma@entry\ZREF@NAME@first 1\else
1637 \ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space
1638 \PackageWarning{zref-marks}{%
1639 Use ‘top’, ‘first’ or ‘bot’ for the list values%
1640 \MessageBreak
1641 in the third argument of \string\zref@marks@register.%
1642 \MessageBreak
1643 Ignoring unkown value ‘\comma@entry’%
1644 }%
1645 \else
1646 \zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}%
1647 \fi
1648 \ltx@gobble
1649 }%
1650 \ltx@gobbletwo
1651 }%
1652 \fi
1653 }
1654 \def\ZREF@STAR{*}
1655 \def\ZREF@NAME@top{top}
```

```

1656 \def\ZREF@NAME@first{first}
1657 \def\ZREF@NAME@bot{bot}
1658 \def\ZREF@MARKS@DefineProp#1{%
1659   \zref@ifpropundefined{#1\ZREF@TempName}{%
1660     \ifnum\ZREF@TempNum=\ltx@zero
1661       \begingroup
1662         \edef\x{\endgroup
1663           \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
1664             \expandafter\noexpand\csname#1mark\endcsname
1665             }%
1666         }%
1667       \x
1668     \else
1669       \begingroup
1670         \edef\x{\endgroup
1671           \noexpand\zref@newprop*{#1\ZREF@TempName}[]{%
1672             \expandafter\noexpand\csname#1marks\endcsname
1673             \ZREF@TempNum
1674             }%
1675         }%
1676       \x
1677     \fi
1678   }{%
1679     \PackageWarning{zref-marks}{%
1680       \string\zref@marks@register\ltx@space does not generate the%
1681       \MessageBreak
1682       new property ‘#1\ZREF@TempName’, because\MessageBreak
1683       it is already defined%
1684     }%
1685   }%
1686 }
1687 </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.

```

1688 <*runs>
1689 \NeedsTeXFormat{LaTeX2e}
1690 \ProvidesPackage{zref-runs}%
1691 [2023-09-14 v2.35 Module runs for zref (HO)]%

```

\zruns

```

1692 \providecommand*\zruns{0}%
1693 \AtBeginDocument{%
1694   \edef\zruns{\number\numexpr\zruns+1}%
1695   \begingroup
1696     \def\on@line{}%
1697     \PackageInfo{zref-runs}{LaTeX runs: \zruns}%
1698     \if@filesw
1699       \immediate\write\@mainaux{%
1700         \string\gdef\string\zruns{\zruns}%
1701       }%
1702     \fi
1703   \endgroup
1704 }
1705 </runs>

```

6.14 Module perpage

```

1706 <*perpage>

```

```

1707 \NeedsTeXFormat{LaTeX2e}
1708 \ProvidesPackage{zref-perpage}%
1709 [2023-09-14 v2.35 Module perpage for zref (HO)]%
1710 \RequirePackage{zref-base}[2019/11/29]
1711 \ifx\ZREF@base@ok Y%
1712 \else
1713 \expandafter\endinput
1714 \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.

```

1715 \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.

```

1716 \ZREF@NewPropPageValue
1717 \zref@newlist{perpage}
1718 \zref@addprops{perpage}{abspage,page,pagevalue}

```

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

```

1719 \newcounter{zpage}

```

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

```

1720 \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`.

```

1721 \let\ZREF@org@stepcounter\stepcounter
1722 \def\stepcounter#1{%
1723   \ifcsname @stepcounterhook@#1\endcsname
1724     \csname @stepcounterhook@#1\endcsname
1725   \fi
1726   \ZREF@org@stepcounter{#1}%
1727 }

```

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>.

```

1728 \IfFormatAtLeastTF{2020/10/01}
1729 {
1730   \AddToHook{package/amstext/after}
1731     {\def\stepcounter#1{%
1732       \iffirstchoice@
1733         \ifcsname @stepcounterhook@#1\endcsname
1734           \csname @stepcounterhook@#1\endcsname
1735         \fi
1736         \addtocounter{#1}\@ne
1737         \begingroup \let\@elt\@stpelt \csname cl@#1\endcsname \endgroup
1738       \fi
1739     }}
1740 }{}
1741 %   \end{macrocode}
1742 %
1743 %
1744 %   \cs{@stpelt} must be adapted due to the change in latex
1745 %   2015-01, see https://github.com/ho-tex/zref/issues/26
1746 %   \begin{macrocode}
1747 \let\ZREF@org@@stpelt\@stpelt
1748 \def\@stpelt#1{%
1749   \ifcsname ZREF@perpage@#1\endcsname
1750     \begingroup
1751     \let\stepcounter\ZREF@org@stepcounter

```

```

1752     \ZREF@org@stpelt{#1}%
1753   \endgroup
1754   \expandafter\ltx@gobbletwo
1755   \fi
1756   \ZREF@org@stpelt{#1}%
1757 }

```

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

```

1758 \ZREF@IfDefinable\zmakeperpage\def{%
1759   {%
1760     \@ifnextchar[\ZREF@makeperpage@opt{\ZREF@@makeperpage[\ltx@zero]}%
1761   }%
1762 }

```

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

```

1763 \def\ZREF@makeperpage@opt[#1]{%
1764   \begingroup
1765     \edef\x{\endgroup
1766       \noexpand\ZREF@@makeperpage[\number\numexpr#1-1\relax]%
1767     }%
1768   \x
1769 }

1770 \def\ZREF@@makeperpage[#1]#2{%
1771   \@ifundefined{@stepcounterhook@#2}{%
1772     \expandafter\gdef\csname @stepcounterhook@#2\endcsname{%}
1773   }{%
1774     \expandafter\gdef\csname ZREF@perpage@#2\endcsname{%
1775       \ZREF@@perpage@step{#2}{#1}%
1776     }%
1777     \expandafter\g@addto@macro\csname @stepcounterhook@#2\endcsname{%
1778       \ifcsname ZREF@perpage@#2\endcsname
1779         \csname ZREF@perpage@#2\endcsname
1780       \fi
1781     }%
1782 }

```

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

```

1783 \def\ZREF@@perpage@step#1#2{%

```

First the reference is generated.

```

1784   \global\advance\c@zref@unique\ltx@one
1785   \begingroup
1786     \expandafter
1787     \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.

```

1788     \zref@ifrefundefined\thezref@unique{%
1789       \global\c@zpage=\c@page
1790       \global\let\thezpage\thepage
1791       \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1792         \number\c@abspage
1793       }%
1794     }{%

```

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

```

1795     \global\c@zpage=\zref@extract\thezref@unique{pagevalue}\relax

```



```

1796     \xdef\thezpage{\noexpand\zref@extract{\thezref@unique}{page}}%
1797     \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1798         \zref@extractdefault\thezref@unique
1799         {abspage}{\number\c@abspage}%
1800     }%
1801 }%

```

Page changes are detected by a changed absolute page number.

```

1802     \expandafter\ifx\csname ZREF@abspage@#1\endcsname
1803         \csname ZREF@currentabspage@#1\endcsname
1804     \else
1805         \global\csname c@#1\endcsname=#2\relax
1806         \global\expandafter\let
1807             \csname ZREF@currentabspage@#1\endcsname
1808             \csname ZREF@abspage@#1\endcsname
1809     \fi
1810 \endgroup
1811 }

```

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

```

1812 \ZREF@ifDefinable\zunmakeperpage\def{%
1813     #1{%
1814         \global\expandafter
1815         \let\csname ZREF@perpage@#1\endcsname\@undefined
1816     }%
1817 }

1818 </perpage>

```

6.15 Module `titleref`

```

1819 <*titleref>
1820 \NeedsTeXFormat{LaTeX2e}
1821 \ProvidesPackage{zref-titleref}%
1822 [2023-09-14 v2.35 Module titleref for zref (HO)]%
1823 \RequirePackage{zref-base}[2019/11/29]
1824 \ifx\ZREF@base@ok Y%
1825 \else
1826     \expandafter\endinput
1827 \fi
1828 \RequirePackage{getttitlestring}[2009/12/08]

```

6.15.1 Implementation

```
1829 \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.

```

1830 \ZREF@NewPropTitle
1831 \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.

```
1832 \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`.

```
1833 \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.

```
1834 \ZREF@Robust\def\zref@titleref@cleanup#1{%
1835   \begingroup
1836   \toks@\expandafter{%
1837     \ZREF@titleref@hook
1838     #1%
1839   }%
1840   \expandafter\endgroup
1841   \expandafter\def\expandafter\ZREF@titleref@hook\expandafter{%
1842     \the\toks@
1843   }%
1844 }%
```

`\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.

```
1845 \newif\ifzref@titleref@stripperperiod
1846 \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.

```
1847 \ZREF@Robust\def\zref@titleref@setcurrent#1{%
1848   \ifzref@titleref@expand
1849     \GetTitleStringExpand{#1}%
1850   \else
1851     \GetTitleStringNonExpand{#1}%
1852   \fi
1853   \edef\zref@titleref@current{%
1854     \detokenize\expandafter{\GetTitleStringResult}%
1855   }%
1856   \ifzref@titleref@stripperperiod
1857     \edef\zref@titleref@current{%
1858       \expandafter\ZREF@stripperperiod\zref@titleref@current
1859       \ltx@empty.\ltx@empty\@nil
1860     }%
1861   \fi
1862 }%
1863 \GetTitleStringDisableCommands{%
1864   \ZREF@titleref@hook
1865 }
```

`\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`.

```
1866 \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`.

```
1867 \define@key{ZREF@TR}{expand}[true]{%
1868   \csname zref@titleref@expand#1\endcsname
1869 }%
```

```

1870 \define@key{ZREF@TR}{stripperperiod}[true]{%
1871   \csname zref@titleref@stripperperiod#1\endcsname
1872 }%
1873 \define@key{ZREF@TR}{cleanup}{%
1874   \zref@titleref@cleanup{#1}%
1875 }%
1876 \define@key{ZREF@TR}{title}{%
1877   \def\zref@titleref@current{#1}%
1878 }%
1879 \ZREF@IfDefinable\ztitlerefsetup\def{%
1880   {\kvsetkeys{ZREF@TR}}%
1881 }%

```

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

```

1882 \ZREF@IfDefinable\ztitleref\def{%
1883   {\zref@wrapper@babel\ZREF@titleref}%
1884 }%
1885 \def\ZREF@titleref#1{%
1886   \begingroup
1887     \zref@refused{#1}%
1888     \let\label\ltx@gobble
1889     \zref@extract{#1}{title}%
1890   \endgroup
1891 }%

```

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.

```

1892 \AtBeginDocument{%
1893   \ZREF@patch{@caption}{%
1894     \long\def\@caption#1[#2]{%
1895       \zref@titleref@setcurrent{#2}%
1896       \ZREF@org@caption{#1}[#2]}%
1897   }%
1898 }%

```

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

```

1899 \ZREF@patch{@part}{%
1900   \def\@part[#1]{%
1901     \zref@titleref@setcurrent{#1}%
1902     \ZREF@org@part[#1]}%
1903   }%
1904 }%
1905 \ZREF@patch{@chapter}{%
1906   \def\@chapter[#1]{%
1907     \zref@titleref@setcurrent{#1}%
1908     \ZREF@org@chapter[#1]}%
1909   }%
1910 }%
1911 \ZREF@patch{@sect}{%
1912   \def\@sect#1#2#3#4#5#6[#7]{%
1913     \zref@titleref@setcurrent{#7}%
1914     \ZREF@org@sect{#1}{#2}{#3}{#4}{#5}{#6}[#7]}%
1915   }%
1916 }%

```

The star versions of the section commands.

```

1917 \ZREF@patch{@spart}{%
1918   \def\@spart#1{%
1919     \zref@titleref@setcurrent{#1}%

```

```

1920     \ZREF@org@@spart{#1}%
1921   }%
1922 }%
1923 \ZREF@patch{@schapter}{%
1924   \def\schapter#1{%
1925     \zref@titleref@setcurrent{#1}%
1926     \ZREF@org@@schapter{#1}%
1927   }%
1928 }%
1929 \ZREF@patch{@ssect}{%
1930   \def\ssect#1#2#3#4#5{%
1931     \zref@titleref@setcurrent{#5}%
1932     \ZREF@org@@ssect{#1}{#2}{#3}{#4}{#5}%
1933   }%
1934 }%

```

6.15.4 Environment description

```

1935 \ZREF@patch{descriptionlabel}{%
1936   \def\descriptionlabel#1{%
1937     \zref@titleref@setcurrent{#1}%
1938     \ZREF@org@descriptionlabel{#1}%
1939   }%
1940 }%

```

6.15.5 Class memoir

```

1941 \@ifclassloaded{memoir}{%
1942   \ltx@ifundefined{ifheadnameref}{}%
1943   \def\@chapter[#1]#2{%
1944     \ltx@ifundefined{ch@pt@c}{%
1945       \zref@titleref@setcurrent{#1}%
1946     }{%
1947       \ifx\ch@pt@c\ltx@empty
1948         \zref@titleref@setcurrent{#2}%
1949       \else
1950         \def\NR@temp{#1}%
1951         \ifx\NR@temp\ltx@empty
1952           \expandafter\zref@titleref@setcurrent
1953           \expandafter{\ch@pt@c}%
1954         \else
1955           \ifheadnameref
1956             \zref@titleref@setcurrent{#1}%
1957           \else
1958             \expandafter\zref@titleref@setcurrent
1959             \expandafter{\ch@pt@c}%
1960           \fi
1961         \fi
1962       \fi
1963     }%
1964     \ZREF@org@@chapter[#{#1}]{#2}%
1965   }%
1966   \ZREF@patch{M@sect}{%
1967     \def\M@sect#1#2#3#4#5#6[#7][#8]{%
1968       \ifheadnameref
1969         \zref@titleref@setcurrent{#8}%
1970       \else
1971         \zref@titleref@setcurrent{#7}%
1972       \fi
1973       \ZREF@org@M@sect{#1}{#2}{#3}{#4}{#5}{#6}[#{#7}][#{#8}]%
1974     }%
1975   }%
1976 }%
1977 }{}%

```

6.15.6 Class beamer

```
1978 \@ifclassloaded{beamer}{%
1979   \ZREF@patch{beamer@section}{%
1980     \long\def\beamer@section[#1]{%
1981       \zref@titleref@setcurrent{#1}%
1982       \ZREF@org@beamer@section[#{1}]%
1983     }%
1984   }%
1985   \ZREF@patch{beamer@subsection}{%
1986     \long\def\beamer@subsection[#1]{%
1987       \zref@titleref@setcurrent{#1}%
1988       \ZREF@org@beamer@subsection[#{1}]%
1989     }%
1990   }%
1991   \ZREF@patch{beamer@subsubsection}{%
1992     \long\def\beamer@subsubsection[#1]{%
1993       \zref@titleref@setcurrent{#1}%
1994       \ZREF@org@beamer@subsubsection[#{1}]%
1995     }%
1996   }%
1997 }{}
```

6.15.7 Package titlesec

```
1998 \@ifpackageloaded{titlesec}{%
1999   \ZREF@patch{ttl@sect@i}{%
2000     \def\ttl@sect@i#1#2[#3]#4{%
2001       \zref@titleref@setcurrent{#4}%
2002       \ZREF@org@ttl@sect@i{#1}#{2}[#{3}]#{4}%
2003     }%
2004   }%
2005   \ZREF@patch{ttl@straight@i}{%
2006     \def\ttl@straight@i#1[#2]#3{%
2007       \def\ZREF@temp{#2}%
2008       \ifx\ZREF@temp\ltx@empty
2009         \zref@titleref@setcurrent{#3}%
2010       \else
2011         \zref@titleref@setcurrent{#2}%
2012       \fi
2013       \ZREF@org@ttl@straight@i{#1}[#{2}]#{3}%
2014     }%
2015   }%
2016 }{}
```

6.15.8 Package longtable

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

```
2017 \@ifpackageloaded{longtable}{%
2018   \ZREF@patch{LT@caption}{%
2019     \def\LT@caption#1[#2]#3{%
2020       \ZREF@org@LT@caption{#1}[#{2}]#{3}%
2021       \zref@titleref@setcurrent{#2}%
2022     }%
2023   }%
2024 }{}
```

6.15.9 Package listings

Package listings: support for its caption.

```
2025 \@ifpackageloaded{listings}{%
2026   \ZREF@patch{lst@MakeCaption}{%
2027     \def\lst@MakeCaption{%
```

```

2028     \ifx\lst@label\ltx@empty
2029     \else
2030     \expandafter\zref@titleref@setcurrent\expandafter{%
2031     \lst@caption
2032     }%
2033     \fi
2034     \ZREF@org@lst@MakeCaption
2035     }%
2036     }%
2037     }-{}%

```

6.15.10 Theorems

```

2038 \ZREF@patch{@opargbegintheorem}{%
2039 \def\@opargbegintheorem#1#2#3{%
2040 \zref@titleref@setcurrent{#3}%
2041 \ZREF@org@@opargbegintheorem{#1}{#2}{#3}%
2042 }%
2043 }%
2044 \@ifpackageloaded{amsthm}{%
2045 \begingroup
2046 \edef\x{macro:\string#1\string#2[\string#3]}%
2047 \@onelevel@sanitize\x
2048 \def\y#1->#2\@nil{#1}%
2049 \edef\z{\expandafter\y\meaning\@begintheorem->\@nil}%
2050 \@onelevel@sanitize\z
2051 \expandafter\endgroup
2052 \ifx\x\z
2053 \ZREF@patch{@begintheorem}{%
2054 \def\@begintheorem#1#2[#3]{%
2055 \zref@titleref@setcurrent{#3}%
2056 \ZREF@org@@begintheorem{#1}{#2}[#3]}%
2057 }%
2058 }%
2059 \fi
2060 }-{}%
2061 }
2062 </titleref>

```

6.16 Module xr

```

2063 <*xr>
2064 \NeedsTeXFormat{LaTeX2e}
2065 \ProvidesPackage{zref-xr}%
2066 [2023-09-14 v2.35 Module xr for zref (HO)]%
2067 \RequirePackage{zref-base}[2019/11/29]
2068 \ifx\ZREF@base@ok Y%
2069 \else
2070 \expandafter\endinput
2071 \fi
2072 \RequirePackage{keyval}
2073 \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.

```

2074 \zref@newprop{url}{}%
2075 \zref@newprop{urluse}{}%
2076 \zref@newprop{externaldocument}{}%

```

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

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

```

2077 \providecommand*\zref@xr@ext}{%
2078 \ltx@ifundefined{XR@ext}{pdf}{\XR@ext}%
2079 }%
```

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

```

2080 \newif\ifZREF@xr@zreflabel

2081 \SetupKeyvalOptions{%
2082 family=ZREF@XR,%
2083 prefix=ZREF@xr%
2084 }
2085 \DeclareBoolOption[true]{tozreflabel}
2086 \DeclareBoolOption[false]{toltxlabel}
2087 \DeclareBoolOption{verbose}
2088 \define@key{ZREF@XR}{ext}{%
2089 \def\zref@xr@{#1}%
2090 }
2091 \DeclareBoolOption[false]{urluse}
```

`\zxrsetup`

```

2092 \newcommand*\zxrsetup}{%
2093 \kvsetkeys{ZREF@XR}%
2094 }%
```

`\ZREF@xr@URL`

```

2095 \newcount\ZREF@xr@URL
2096 \ZREF@xr@URL=\ltx@zero
```

`\ZREF@xr@AddURL`

```

2097 \def\ZREF@xr@AddURL#1{%
2098 \begingroup
2099 \def\ZREF@temp{#1}%
2100 \count@=\ltx@one
2101 \ZREF@xr@@AddUrl
2102 \endgroup
2103 }
```

`\ZREF@xr@@AddUrl`

```

2104 \def\ZREF@xr@@AddUrl{%
2105 \ifnum\count@>\ZREF@xr@URL
2106 \global\advance\ZREF@xr@URL by\ltx@one
2107 \xdef\ZREF@xr@theURL{\romannumeral\ZREF@xr@URL}%
2108 \global\expandafter\let
2109 \csname Z@U@\ZREF@xr@theURL\endcsname\ZREF@temp
2110 \@PackageInfo{zref-xr}{%
2111 \ltx@backslashchar Z@U@\ZREF@xr@theURL:\MessageBreak
2112 \ZREF@temp\MessageBreak
2113 }%
2114 \else
2115 \expandafter
2116 \ifx\csname Z@U@\romannumeral\count@\endcsname\ZREF@temp
2117 \xdef\ZREF@xr@theURL{\romannumeral\count@}%
2118 \else
2119 \expandafter\expandafter\expandafter\ZREF@xr@@AddUrl
2120 \fi
2121 \fi
2122 }
```

`\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).

```

2123 \ZREF@ifDefinable\zexternaldocument\def{%
2124   {%
2125     \ZREF@NewPropAnchor
2126     \ZREF@NewPropTitle
2127     \begingroup
2128     \csname @safe@actives@true\endcsname
2129     \makeatletter
2130     \@ifstar{%
2131       \ZREF@xr@zreflabelfalse
2132       \@testopt\ZREF@xr@externaldocument{}%
2133     }{%
2134       \ZREF@xr@zreflabeltrue
2135       \@testopt\ZREF@xr@externaldocument{}%
2136     }%
2137   }%
2138 }%

```

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

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

```

2139 \def\ZREF@xr@externaldocument[#1]#2{%
2140   \def\ZREF@xr@prefix{#1}%
2141   \let\ZREF@xr@filelist\ltx@empty
2142   \edef\ZREF@xr@externalfile{#2}%
2143   \edef\ZREF@xr@file{\ZREF@xr@externalfile.aux}%
2144   \filename@parse{#2}%
2145   \@testopt\ZREF@xr@graburl{#2.\zref@xr@ext}%
2146 }%
2147 \def\ZREF@xr@graburl[#1]{%
2148   \edef\ZREF@xr@url{#1}%
2149   \ifZREF@xr@urluse
2150     \expandafter\ZREF@xr@AddURL\expandafter{\ZREF@xr@url}%
2151     \expandafter\def\expandafter\ZREF@xr@url
2152     \expandafter{\csname Z@U@\ZREF@xr@theURL\endcsname}%
2153   \fi
2154   \ZREF@xr@checkfile
2155 \endgroup
2156 }%

```

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

```

2157 \def\ZREF@xr@checkfile{%
2158   \openin\@inputcheck\ZREF@xr@file\relax
2159   \ifeof\@inputcheck
2160     \PackageWarning{zref-xr}{%
2161       File '\ZREF@xr@file' not found or empty,\MessageBreak
2162       labels not imported%
2163     }%
2164   \else
2165     \PackageInfo{zref-xr}{%
2166       Label \ifZREF@xr@zreflabel (zref) \fi
2167       import from '\ZREF@xr@file'%
2168     }%
2169     \def\ZREF@xr@found{0}%
2170     \def\ZREF@xr@ignored@empty{0}%
2171     \def\ZREF@xr@ignored@zref{0}%

```



```

2172 \def\ZREF@xr@ignored@ltx{0}%
2173 \ZREF@xr@processfile
2174 \closein\@inputcheck
2175 \begingroup
2176 \let\on@line\ltx@empty
2177 \PackageInfo{zref-xr}{%
2178   Statistics for '\ZREF@xr@file':\MessageBreak
2179   \ZREF@xr@found\space
2180   \ifZREF@xr@zreflabel zref\else LaTeX\fi\space
2181   label(s) found%
2182   \ifnum\ZREF@xr@ignored@empty>0 %
2183     ,\MessageBreak
2184     \ZREF@xr@ignored@empty\space empty label(s) ignored%
2185   \fi
2186   \ifnum\ZREF@xr@ignored@zref>0 %
2187     ,\MessageBreak
2188     \ZREF@xr@ignored@zref\space
2189     duplicated zref label(s) ignored%
2190   \fi
2191   \ifnum\ZREF@xr@ignored@ltx>0 %
2192     ,\MessageBreak
2193     \ZREF@xr@ignored@ltx\space
2194     duplicated latex label(s) ignored%
2195   \fi
2196 }%
2197 \endgroup
2198 \fi
2199 \ifx\ZREF@xr@filelist\ltx@empty
2200 \else
2201 \edef\ZREF@xr@file{%
2202 \expandafter\ltx@car\ZREF@xr@filelist\@nil
2203 }%
2204 \edef\ZREF@xr@filelist{%
2205 \expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil
2206 }%
2207 \expandafter\ZREF@xr@checkfile
2208 \fi
2209 }%

```

\ZREF@xr@processfile

```

2210 \def\ZREF@xr@processfile{%
2211 \read\@inputcheck to\ZREF@xr@line
2212 \expandafter\ZREF@xr@processline\ZREF@xr@line..\ZREF@nil
2213 \ifeof\@inputcheck
2214 \else
2215 \expandafter\ZREF@xr@processfile
2216 \fi
2217 }%

```

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

```

2218 \long\def\ZREF@xr@processline#1#2#3\ZREF@nil{%
2219 \def\x{#1}%
2220 \toks@{#2}%
2221 \ifZREF@xr@zreflabel
2222 \ifx\x\ZREF@xr@zref@newlabel
2223 \expandafter
2224 \ZREF@xr@process@zreflabel\ZREF@xr@line...\ZREF@nil
2225 \fi
2226 \else
2227 \ifx\x\ZREF@xr@newlabel
2228 \expandafter
2229 \ZREF@xr@process@label\ZREF@xr@line... []\ZREF@nil
2230 \fi

```



```

2293 \ifx\#4\%
2294 \else
2295 % ntheorem knows an optional argument at the end of \newlabel
2296 \ZREF@NewPropTheotype
2297 \expandafter\ltx@LocalAppendToMacro
2298 \csname\ZREF@xr@temprefname\endcsname{\thetype{#4}}%
2299 \fi
2300 \expandafter\ltx@LocalAppendToMacro
2301 \csname\ZREF@xr@temprefname\expandafter\endcsname\expandafter{%
2302 \expandafter\externaldocument\expandafter{%
2303 \ZREF@xr@externalfile
2304 }%
2305 }%
2306 \ZREF@xr@urlcheck\ZREF@xr@tempname
2307 \ifZREF@xr@tozreflabel
2308 \@ifundefined{\ZREF@xr@refname}{%
2309 \ifZREF@xr@verbose
2310 \PackageInfo{zref-xr}{%
2311 Import to zref label '\ZREF@xr@prefix#1'%
2312 }%
2313 \fi
2314 \global\expandafter
2315 \let\csname\ZREF@xr@refname\expandafter\endcsname
2316 \csname\ZREF@xr@temprefname\endcsname
2317 }{%
2318 \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2319 }%
2320 \fi
2321 \ifZREF@xr@toltxlabel
2322 \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2323 \fi
2324 }
2325 \def\ZREF@xr@zref@newlabel{\zref@newlabel}%
2326 \def\ZREF@xr@newlabel{\newlabel}%
2327 \def\ZREF@xr@@input{\@input}%
2328 \def\ZREF@xr@relax{\relax}%

```

\ZREF@xr@tolabel

```

2329 \def\ZREF@xr@tolabel#1#2{%
2330 \ifZREF@xr@verbose
2331 \PackageInfo{zref-xr}{%
2332 Import to LaTeX label '#2'%
2333 }%
2334 \fi
2335 \zref@wrapper@unexpanded{%
2336 \expandafter\xdef\csname r@#2\endcsname{%
2337 {%
2338 \ltx@ifundefined{M@TitleReference}{%
2339 \ltx@ifundefined{TR@TitleReference}{%
2340 \zref@extractdefault{#1}{default}{}%
2341 }{%
2342 \noexpand\TR@TitleReference
2343 {\zref@extractdefault{#1}{default}{}}%
2344 {\zref@extractdefault{#1}{title}{}}%
2345 }%
2346 }{%
2347 \noexpand\M@TitleReference
2348 {\zref@extractdefault{#1}{default}{}}%
2349 {\zref@extractdefault{#1}{title}{}}%
2350 }%
2351 }%
2352 {\zref@extractdefault{#1}{page}{}}%
2353 \ltx@ifpackageloaded{nameref}{%

```

```

2354     {\zref@extractdefault{#1}{title}{}}%
2355     {\zref@extractdefault{#1}{anchor}{}}%
2356     \zref@ifrefcontainsprop{#1}{urluse}{%
2357       {\zref@extractdefault{#1}{urluse}{}}%
2358     }{%
2359       {\zref@extractdefault{#1}{url}{}}%
2360     }%
2361   }{}%
2362 }%
2363 }%
2364 }

```

\ZREF@xr@zref@ignorewarning

```

2365 \def\ZREF@xr@zref@ignorewarning#1{%
2366   \PackageWarningNoLine{zref-xr}{%
2367     Zref label ‘#1’ is already in use\MessageBreak
2368     in file ‘\ZREF@xr@file’%
2369   }%
2370   \edef\ZREF@xr@ignored@zref{%
2371     \the\numexpr\ZREF@xr@ignored@zref+1%
2372   }%
2373 }%

```

\ZREF@xr@ltx@ignorewarning

```

2374 \def\ZREF@xr@ltx@ignorewarning#1{%
2375   \PackageWarningNoLine{zref-xr}{%
2376     LaTeX label ‘#1’ is already in use\MessageBreak
2377     in file ‘\ZREF@xr@file’%
2378   }%
2379   \edef\ZREF@xr@ignored@ltx{%
2380     \the\numexpr\ZREF@xr@ignored@ltx+1%
2381   }%
2382 }%

```

\ZREF@xr@checklist

```

2383 \def\ZREF@xr@checklist#1#2#3\ZREF@nil{%
2384   \ifx\@undefined#1\relax
2385     \expandafter\ZREF@xr@checkkey\string#1\@nil
2386   \fi
2387   \ifx\#3\%
2388   \else
2389     \ltx@ReturnAfterFi{%
2390       \ZREF@xr@checklist#3\ZREF@nil
2391     }%
2392   \fi
2393 }%
2394 \def\ZREF@xr@checkkey#1#2\@nil{%
2395   \zref@ifpropundefined{#2}{%
2396     \zref@newprop{#2}{}%
2397   }{}%
2398 }%

```

\ZREF@xr@scanparams

```

2399 \def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{%
2400   \let#1\ltx@empty
2401   \ZREF@foundfalse
2402   \ZREF@xr@scantitleref#1#2\TR@TitleReference{}{}\ZREF@nil
2403   \ifZREF@found
2404   \else
2405     \ltx@LocalAppendToMacro#1{\default{#2}}%
2406   \fi
2407   % page

```

```

2408 \ltx@LocalAppendToMacro#1{\page{#3}}%
2409 % nameref title
2410 \ifZREF@found
2411 \else
2412   \ifx\\#4\\%
2413   \else
2414     \def\ZREF@xr@temp{#4}%
2415     \ifx\ZREF@xr@temp\ZREF@xr@relax
2416     \else
2417       \ltx@LocalAppendToMacro#1{\title{#4}}%
2418     \fi
2419   \fi
2420 \fi
2421 % anchor
2422 \ifx\\#5\\%
2423 \else
2424   \ltx@LocalAppendToMacro#1{\anchor{#5}}%
2425 \fi
2426 \ifx\\#6\\%
2427 \else
2428   \ifZREF@xr@urluse
2429     \ZREF@xr@AddURL{#6}%
2430     \expandafter\ltx@LocalAppendToMacro\expandafter#1%
2431     \expandafter{%
2432       \expandafter\urluse\expandafter{%
2433         \csname Z@U@\ZREF@xr@theURL\endcsname
2434       }%
2435     }%
2436   \else
2437     \ltx@LocalAppendToMacro#1{\url{#6}}%
2438   \fi
2439 \fi
2440 }%

```

\ZREF@xr@scantitleref

```

2441 \def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{%
2442   \ifx\\#5\\%
2443   \else
2444     \ltx@LocalAppendToMacro#1{%
2445       \default{#3}%
2446       \title{#4}%
2447     }%
2448     \ZREF@foundtrue
2449   \fi
2450 }%

```

\ZREF@xr@urlcheck

```

2451 \def\ZREF@xr@urlcheck#1{%
2452   \zref@ifrefcontainsprop{#1}{anchor}{%
2453     \zref@ifrefcontainsprop{#1}{url}{%
2454       }{%
2455         \expandafter
2456         \ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
2457         \expandafter{%
2458           \csname url\ifZREF@xr@urluse use\fi
2459           \expandafter\endcsname\expandafter{\ZREF@xr@url}%
2460         }%
2461       }%
2462     }{%
2463     }%
2464 }%
2465 \</xr>

```

6.17 Module `hyperref`

```
UNFINISHED :-(  
2466 <*hyperref>  
2467 \NeedsTeXFormat{LaTeX2e}  
2468 \ProvidesPackage{zref-hyperref}%  
2469 [2023-09-14 v2.35 Module hyperref for zref (H0)]%  
2470 \RequirePackage{zref-base}[2019/11/29]  
2471 \ifx\ZREF@base@ok Y%  
2472 \else  
2473 \expandafter\endinput  
2474 \fi  
  
2475 \ZREF@NewPropAnchor  
2476 \zref@addprop\ZREF@mainlist{anchor}%  
2477 </hyperref>
```

6.18 Module `savepos`

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

6.18.1 Identification

```
2478 <*savepos>  
2479 \NeedsTeXFormat{LaTeX2e}  
2480 \ProvidesPackage{zref-savepos}%  
2481 [2023-09-14 v2.35 Module savepos for zref (H0)]%  
2482 \RequirePackage{zref-base}[2019/11/29]  
2483 \ifx\ZREF@base@ok Y%  
2484 \else  
2485 \expandafter\endinput  
2486 \fi
```

6.18.2 Availability

First we check, whether the feature is available.

```
2487 \ifx\directlua\undefined  
2488 \ltx@ifundefined{pdfsavepos}{%  
2489 \PackageError\ZREF@name{%  
2490 \string\pdfsavepos\space is not supported.\MessageBreak  
2491 It is provided by pdfTeX (1.40) or XeTeX%  
2492 }\ZREF@UpdatePdfTeX  
2493 \endinput  
2494 }{}%  
2495 \fi
```

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

```
2496 \ifpdf  
2497 \else  
2498 \ltx@ifundefined{pdftexversion}{%  
2499 }{%  
2500 \ifnum\pdftexversion<140 %  
2501 \PackageError\ZREF@name{%  
2502 \string\pdfsavepos\space is not supported in DVI mode%  
2503 \MessageBreak  
2504 of this pdfTeX version%  
2505 }\ZREF@UpdatePdfTeX  
2506 \expandafter\expandafter\expandafter\endinput  
2507 \fi  
2508 }%  
2509 \fi
```

6.18.3 Setup

```
2510 \zref@newlist{savepos}
2511 \ifx\directlua\@undefined
2512 \zref@newprop*{posx}[0]{\the\pdflastxpos}
2513 \zref@newprop*{posy}[0]{\the\pdflastypos}
2514 \else
2515 \zref@newprop*{posx}[0]{\the\lastxpos}
2516 \zref@newprop*{posy}[0]{\the\lastypos}
2517 \fi
2518 \zref@addprops{savepos}{posx,posy}
```

6.18.4 User macros

`\zref@savepos`

```
2519 \ifx\directlua\@undefined
2520 \def\zref@savepos{%
2521   \if@filesw
2522     \pdfsavepos
2523   \fi
2524 }
2525 \else
2526 \def\zref@savepos{%
2527   \if@filesw
2528     \savepos
2529   \fi
2530 }
2531 \fi
```

`\ZREF@zsavepos`

```
2532 \def\ZREF@zsavepos#1#2#3{%
2533   \@bsphack
2534   \if@filesw
2535     \zref@savepos
2536     #1{#3}{#2}%
2537     \ltx@ifundefined{TeXeTstate}{%
2538     }{%
2539     \ifnum\TeXeTstate=\ltx@zero
2540     \else
2541     \zref@savepos
2542     \fi
2543     }%
2544   \fi
2545   \@esphack
2546 }
```

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

```
2547 \ZREF@ifDefinable\zsavepos\def{%
2548   {%
2549     \ZREF@zsavepos\zref@labelbylist{savepos}%
2550   }%
2551 }
```

`\zsaveposx`

```
2552 \ZREF@ifDefinable\zsaveposx\def{%
2553   {%
2554     \ZREF@zsavepos\zref@labelbyprops{posx}%
2555   }%
2556 }
```

`\zsaveposy`

```
2557 \ZREF@ifDefinable\zsaveposy\def{%
2558   {%
```

```

2559   \ZREF@zsavepos\zref@labelbyprops{posy}%
2560   }%
2561 }

```

`\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.

```

2562 \newcommand*{\zposx}[1]{%
2563   \zref@extract{#1}{posx}%
2564 }%
2565 \newcommand*{\zposy}[1]{%
2566   \zref@extract{#1}{posy}%
2567 }%

```

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`

```

2568 \let\ZREF@savepos@ok=Y
2569 </savepos>

```

6.19 Module `abspos`

6.19.1 Identification

```

2570 <*abspos>
2571 \NeedsTeXFormat{LaTeX2e}
2572 \ProvidesPackage{zref-abspos}%
2573 [2023-09-14 v2.35 Module abspos for zref (H0)]%
2574 \RequirePackage{zref-base}[2019/11/29]
2575 \ifx\ZREF@base@ok Y%
2576 \else
2577   \expandafter\endinput
2578 \fi
2579 \RequirePackage{zref-savepos}[2019/11/29]
2580 \ifx\ZREF@savepos@ok Y%
2581 \else
2582   \expandafter\endinput
2583 \fi
2584 \RequirePackage{zref-pagelayout}[2019/11/29]
2585 \zref@addprop{savepos}{abspage}
2586 \zref@addprop{savepos}{pagevalue}

```

`\zref@absposx`

```

2587 \newcommand*{\zref@absposx}[3]{%
2588   \number
2589   \expandafter\zref@absposnumx\expandafter{%
2590     \number\zref@extractdefault{#1}{abspage}{0}%
2591     }{#2}{#3}%
2592   \ltx@space
2593 }

```

`\zref@absposy`

```

2594 \newcommand*{\zref@absposy}[3]{%
2595   \number
2596   \expandafter\zref@absposnumy\expandafter{%
2597     \number\zref@extractdefault{#1}{abspage}{0}%
2598     }{#2}{#3}%

```



```

2599 \ltx@space
2600 }

\zref@absposnumx
2601 \newcommand*{\zref@absposnumx}[3]{%
2602 \number
2603 % \ifnum#1>\ltx@zero
2604 % \zref@ifrefundefined{thepage#1}{%
2605 % 0%
2606 % }{%
2607 \numexpr\ZREF@absposnum{thepage#1}{#2}{x}{#3}\relax
2608 % }%
2609 % \else
2610 % 0%
2611 % \fi
2612 }

\zref@absposnumy
2613 \newcommand*{\zref@absposnumy}[3]{%
2614 \number
2615 % \ifnum#1>\ltx@zero
2616 % \zref@ifrefundefined{thepage#1}{%
2617 % 0%
2618 % }{%
2619 \numexpr\ZREF@absposnum{thepage#1}{#2}{y}{#3}\relax
2620 % }%
2621 % \else
2622 % 0%
2623 % \fi
2624 }

\ZREF@absposnum
2625 \def\ZREF@absposnum#1#2#3#4{%
2626 \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
2627 0%
2628 }{%
2629 \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
2630 }%
2631 }

\zref@def@absposx
2632 \ZREF@Robust\def\zref@def@absposx#1{%
2633 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposx}%
2634 }

\zref@def@absposy
2635 \ZREF@Robust\def\zref@def@absposy#1{%
2636 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
2637 }

\zref@def@absposnumx
2638 \ZREF@Robust\def\zref@def@absposnumx#1{%
2639 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumx}%
2640 }

\zref@def@absposnumy
2641 \ZREF@Robust\def\zref@def@absposnumy#1{%
2642 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumy}%
2643 }

```

```

\ZREF@def@abspos
2644 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2645   \edef#1{#2{#3}{#4}{#5}}%
2646 }

\zref@absposused
2647 \ZREF@Robust\def\zref@absposused{%
2648   \zref@wrapper@babel\ZREF@abspos@used
2649 }

\ZREF@abspos@used
2650 \def\ZREF@abspos@used#1{%
2651   \zref@refused{#1}%
2652   \zref@ifrefundefined{#1}{%
2653     }{%
2654     \begingroup
2655       \edef\ZREF@temp{%
2656         \zref@extractdefault{#1}{abspage}{0}}%
2657     }%
2658     \ifnum\ZREF@temp>\ltx@zero
2659       \zref@refused{thepage\ZREF@temp}%
2660     \else
2661       \@PackageError{zref-abspos}{%
2662         \string\zref@pos@label@used\ltx@space
2663         needs property ‘abspage’\MessageBreak
2664         in label ‘#1’%
2665       }\@ehc
2666     \fi
2667   \endgroup
2668 }%
2669 }

\zref@absposnumused
2670 \newcommand*{\zref@absposnumused}[1]{%
2671   \ifnum#1>\ltx@zero
2672     \zref@refused{thepage\number#1}%
2673   \else
2674     \@PackageError{zref-abspos}{%
2675       Invalid absolute page number (#1)\MessageBreak
2676       for \string\zref@pos@num@used.\MessageBreak
2677       A positive integer is expected%
2678     }\@ehc
2679   \fi
2680 }

\zref@ifabsposundefined
2681 \def\zref@ifabsposundefined#1{%
2682   \zref@ifrefundefined{#1}\ltx@firsttwo{%
2683     \expandafter\zref@ifabsposnumundefined\expandafter{%
2684       \number\zref@extractdefault{#1}{abspage}{0}}%
2685     }%
2686   }%
2687 }

\zref@ifabsposnumundefined
2688 \def\zref@ifabsposnumundefined#1{%
2689   \ifnum\ZREF@number{#1}>\ltx@zero
2690     \zref@ifrefundefined{thepage#1}%
2691     \ltx@firstoftwo\ltx@secondoftwo
2692   \else
2693     \expandafter\ltx@firstoftwo
2694   \fi
2695 }

```

6.19.2 Media

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

```
2696 \@ifundefined{stockwidth}{\newdimen\stockwidth}{}  
2697 \@ifundefined{stockheight}{\newdimen\stockheight}{}
```

```
\ZREF@abspos@media@width
```

```
2698 \edef\ZREF@abspos@media@width{%  
2699   \ltx@ifundefined{pdfpagewidth}{%  
2700     \ltx@ifundefined{mediawidth}{%  
2701       \ifdim\stockwidth > \z@  
2702         stockwidth%  
2703       \else  
2704         paperwidth%  
2705       \fi  
2706     }{%  
2707       mediawidth%  
2708     }%  
2709   }{%  
2710     pdfpagewidth%  
2711   }%  
2712 }  
2713 \ifluatex  
2714 \def\ZREF@abspos@media@width{pdfpagewidth}%  
2715 \fi
```

```
\ZREF@abspos@media@height
```

```
2716 \edef\ZREF@abspos@media@height{%  
2717   \ltx@ifundefined{pdfpageheight}{%  
2718     \ltx@ifundefined{mediaheight}{%  
2719       \ifdim\stockwidth > \z@  
2720         stockheight%  
2721       \else  
2722         paperheight%  
2723       \fi  
2724     }{%  
2725       mediaheight%  
2726     }%  
2727   }{%  
2728     \noexpand\ifcase\pdfpageheight  
2729       \ifdim\stockwidth > \z@  
2730         stockheight%  
2731       \else  
2732         paperheight%  
2733       \fi  
2734     \noexpand\else  
2735       pdfpageheight%  
2736     \noexpand\fi  
2737   }%  
2738 }  
2739 \ifluatex  
2740 \edef\ZREF@abspos@media@height{%  
2741   \noexpand\ifcase\pageheight  
2742     \ifdim\stockwidth > \z@  
2743       stockheight%  
2744     \else  
2745       paperheight%  
2746     \fi  
2747   \noexpand\else  
2748     pdfpageheight%  
2749   \noexpand\fi}%  
2750 \fi
```

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

```
2751 \def\ZREF@abspos@media@x@left#1{%  
2752   0%  
2753 }
```

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

```
2754 \def\ZREF@abspos@media@x@right#1{%  
2755   \zref@extract{#1}\ZREF@abspos@media@width  
2756 }
```

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

```
2757 \def\ZREF@abspos@media@x@center#1{%  
2758   \ZREF@abspos@media@x@left{#1}%  
2759   +\zref@extract{#1}\ZREF@abspos@media@width/2%  
2760 }
```

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

```
2761 \def\ZREF@abspos@media@y@top#1{%  
2762   \zref@extract{#1}\ZREF@abspos@media@height  
2763 }
```

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

```
2764 \def\ZREF@abspos@media@y@bottom#1{%  
2765   0%  
2766 }
```

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

```
2767 \def\ZREF@abspos@media@y@center#1{%  
2768   \zref@extract{#1}\ZREF@abspos@media@height/2%  
2769 }
```

6.19.3 Paper

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

```
2770 \def\ZREF@abspos@paper@x@left#1{%  
2771   0%  
2772 }
```

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

```
2773 \def\ZREF@abspos@paper@x@right#1{%  
2774   \zref@extract{#1}{paperwidth}%  
2775 }
```

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

```
2776 \def\ZREF@abspos@paper@x@center#1{%  
2777   \zref@extract{#1}{paperwidth}/2%  
2778 }
```

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

```
2779 \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top
```

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

```
2780 \def\ZREF@abspos@paper@y@bottom#1{%  
2781   \ZREF@abspos@paper@y@top{#1}%  
2782   -\zref@extract{#1}{paperheight}%  
2783 }
```

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

```
2784 \def\ZREF@abspos@paper@y@center#1{%  
2785   \ZREF@abspos@paper@y@top{#1}%  
2786   -\zref@extract{#1}{paperheight}/2%  
2787 }
```

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
2788 \let\ZREF@temp\ltx@two
2789 \ltx@ifundefined{pdfhorigin}{}%
2790 % \ifpdf
2791 \let\ZREF@temp\ltx@zero
2792 % \fi
2793 }
2794 \ifluatex
2795 % \ifpdf
2796 \let\ZREF@temp\ltx@zero
2797 % \fi
2798 \fi
2799
2800 \ifx\ZREF@temp\ltx@two
2801 \ifnum\mag=1000 %
2802 \let\ZREF@temp\ltx@one
2803 \fi
2804 \fi
2805 \ifcase\ZREF@temp
2806 \def\ZREF@abspos@origin@x#1{%
2807 \zref@extract{#1}{pdfhorigin}%
2808 }%
2809 \or
2810 \def\ZREF@abspos@origin@x#1{%
2811 4736286%
2812 }%
2813 \or
2814 \def\ZREF@abspos@origin@x#1{%
2815 \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2816 }%
2817 \fi

\ZREF@abspos@origin@y
2818 \let\ZREF@temp\ltx@two
2819 \ltx@ifundefined{pdfvorigin}{}%
2820 % \ifpdf
2821 \let\ZREF@temp\ltx@zero
2822 % \fi
2823 }
2824 \ifluatex
2825 % \ifpdf
2826 \let\ZREF@temp\ltx@zero
2827 % \fi
2828 \fi
2829 \ifx\ZREF@temp\ltx@two
2830 \ifnum\mag=1000 %
2831 \let\ZREF@temp\ltx@one
2832 \fi
2833 \fi
2834 \ifcase\ZREF@temp
2835 \def\ZREF@abspos@origin@y#1{%
2836 \zref@extract{#1}{pdfvorigin}%
2837 }%
2838 \or
2839 \def\ZREF@abspos@origin@y#1{%
2840 4736286%
2841 }%
2842 \or
```

```

2843 \def\ZREF@abspos@origin@y#1{%
2844   \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2845   }%
2846 \fi

```

6.19.5 Header

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

```

2847 \def\ZREF@abspos@head@x@left#1{%
2848   \ZREF@abspos@paper@x@left{#1}%
2849   +\ZREF@abspos@origin@x{#1}%
2850   +\zref@extract{#1}{hoffset}%
2851   +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2852     \zref@extract{#1}{oddsidemargin}%
2853   \else
2854     \zref@extract{#1}{evensidemargin}%
2855   \fi
2856 }

```

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

```

2857 \def\ZREF@abspos@head@x@right#1{%
2858   \ZREF@abspos@head@x@left{#1}%
2859   +\zref@extract{#1}{textwidth}%
2860 }

```

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

```

2861 \def\ZREF@abspos@head@x@center#1{%
2862   \ZREF@abspos@head@x@left{#1}%
2863   +\zref@extract{#1}{textwidth}/2%
2864 }

```

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

```

2865 \def\ZREF@abspos@head@y@top#1{%
2866   \ZREF@abspos@paper@y@top{#1}%
2867   -\ZREF@abspos@origin@y{#1}%
2868   -\zref@extract{#1}{voffset}%
2869   -\zref@extract{#1}{topmargin}%
2870 }

```

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

```

2871 \def\ZREF@abspos@head@y@bottom#1{%
2872   \ZREF@abspos@head@y@top{#1}%
2873   -\zref@extract{#1}{headheight}%
2874 }

```

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

```

2875 \def\ZREF@abspos@head@y@center#1{%
2876   \ZREF@abspos@head@y@top{#1}%
2877   -\zref@extract{#1}{headheight}/2%
2878 }

```

6.19.6 Body

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

```

2879 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left

```

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

```

2880 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right

```

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

```

2881 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center

```

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

```
2882 \def \ZREF@abspos@body@y@top#1{%
2883   \ZREF@abspos@head@y@bottom{#1}%
2884   -\zref@extract{#1}{headsep}%
2885 }
```

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

```
2886 \def \ZREF@abspos@body@y@bottom#1{%
2887   \ZREF@abspos@body@y@top{#1}%
2888   -\zref@extract{#1}{textheight}%
2889 }
```

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

```
2890 \def \ZREF@abspos@body@y@center#1{%
2891   \ZREF@abspos@body@y@top{#1}%
2892   -\zref@extract{#1}{textheight}/2%
2893 }
```

6.19.7 Footer

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

```
2894 \let \ZREF@abspos@foot@x@left \ZREF@abspos@head@x@left
```

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

```
2895 \let \ZREF@abspos@foot@x@right \ZREF@abspos@head@x@right
```

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

```
2896 \let \ZREF@abspos@foot@x@center \ZREF@abspos@head@x@center
```

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

```
2897 \def \ZREF@abspos@foot@y@bottom#1{%
2898   \ZREF@abspos@body@y@bottom{#1}%
2899   -\zref@extract{#1}{footskip}%
2900 }
```

6.19.8 Marginal notes

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

```
2901 \def \ZREF@abspos@marginpar@x@left#1{%
2902   \ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2903     \ZREF@abspos@body@x@right{#1}%
2904     +\zref@extract{#1}{marginparsep}%
2905   \else
2906     \ZREF@abspos@body@x@left{#1}%
2907     -\zref@extract{#1}{marginparsep}%
2908     -\zref@extract{#1}{marginparwidth}%
2909   \fi
2910 }
```

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

```
2911 \def \ZREF@abspos@marginpar@x@right#1{%
2912   \ZREF@abspos@marginpar@x@left{#1}%
2913   +\zref@extract{#1}{marginparwidth}%
2914 }
```

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

```
2915 \def \ZREF@abspos@marginpar@x@center#1{%
2916   \ZREF@abspos@marginpar@x@left{#1}%
2917   +\zref@extract{#1}{marginparwidth}/2%
2918 }
```

```

\ZREF@abspos@marginpar@y@top
2919 \let\ZREF@abspos@marginpar@y@top\ZREF@abspos@body@y@top
ZREF@abspos@marginpar@y@bottom
2920 \let\ZREF@abspos@marginpar@y@bottom\ZREF@abspos@body@y@bottom
ZREF@abspos@marginpar@y@center
2921 \let\ZREF@abspos@marginpar@y@center\ZREF@abspos@body@y@center

```

6.19.9 Stock paper

```

\ZREF@abspos@stock@x@left
2922 \let\ZREF@abspos@stock@x@left\ZREF@abspos@paper@x@left

\ZREF@abspos@stock@x@right
2923 \let\ZREF@abspos@stock@x@right\ZREF@abspos@paper@x@right

\ZREF@abspos@stock@x@center
2924 \let\ZREF@abspos@stock@x@center\ZREF@abspos@paper@x@center

\ZREF@abspos@stock@y@top
2925 \let\ZREF@abspos@stock@y@top\ZREF@abspos@paper@y@top

\ZREF@abspos@stock@y@bottom
2926 \let\ZREF@abspos@stock@y@bottom\ZREF@abspos@paper@y@bottom

\ZREF@abspos@stock@y@center
2927 \let\ZREF@abspos@stock@y@center\ZREF@abspos@paper@y@center
2928 </abspos>

```

6.20 Module dotfill

```

2929 <*.dotfill>
2930 \NeedsTeXFormat{LaTeX2e}
2931 \ProvidesPackage{zref-dotfill}%
2932 [2023-09-14 v2.35 Module dotfill for zref (HO)]%
2933 \RequirePackage{zref-base}[2019/11/29]
2934 \ifx\ZREF@base@ok Y%
2935 \else
2936 \expandafter\endinput
2937 \fi

```

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

```
2938 \RequirePackage{zref-savepos}[2019/11/29]
```

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

```
2939 \zref@require@unique
```

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

```
2940 \RequirePackage{keyval}
```

The definitions of the keys follow.

```

2941 \define@key{ZREF@DF}{unit}{%
2942   \def\ZREF@df@unit{#1}%
2943 }
2944 \define@key{ZREF@DF}{min}{%
2945   \def\ZREF@df@min{#1}%
2946 }
2947 \define@key{ZREF@DF}{dot}{%
2948   \def\ZREF@df@dot{#1}%

```



```

2949 }
Defaults are set, see user interface.
2950 \providecommand\ZREF@df@min{2}
2951 \providecommand\ZREF@df@unit{.44em}
2952 \providecommand\ZREF@df@dot{.}

\zdotfillsetup Configuration of \zdotfill is done by \zdotfillsetup.
2953 \newcommand*{\zdotfillsetup}{\kvsetkeys{ZREF@DF}}

\zdotfill \zdotfill sets labels at the left and the right to get the horizontal position.
\zsavepos is not used, because we do not need the vertical position.
2954 \ZREF@IfDefinable\zdotfill\def{%
2955   {%
2956     \leavevmode
2957     \global\advance\c@zref@unique\ltx@one
2958     \begingroup
2959     \def\ZREF@temp{zref@\number\c@zref@unique}%
2960     \pdfsavepos
2961     \zref@labelbyprops{\thezref@unique L}{posx}%
2962     \setlength{\dimen@}{\ZREF@df@unit}%
2963     \zref@ifrefundefined{\thezref@unique R}{%
2964       \ZREF@dotfill
2965     }{%
2966       \ifnum\numexpr\zposx{\thezref@unique R}%
2967         -\zposx{\thezref@unique L}\relax
2968         <\dimexpr\ZREF@df@min\dimen@\relax
2969       \hfill
2970     \else
2971       \ZREF@dotfill
2972     \fi
2973   }%
2974   \pdfsavepos
2975   \zref@labelbyprops{\thezref@unique R}{posx}%
2976 \endgroup
2977 \kern\z@
2978 }%
2979 }

```

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

```

2980 \def\ZREF@dotfill{%
2981   \cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill
2982 }

2983 </dotfill>

```

6.21 Module `env`

```

2984 <*env>
2985 \NeedsTeXFormat{LaTeX2e}
2986 \ProvidesPackage{zref-env}%
2987 [2023-09-14 v2.35 Module env for zref (HO)]%
2988 \RequirePackage{zref-base}[2019/11/29]
2989 \ifx\ZREF@base@ok Y%
2990 \else
2991   \expandafter\endinput
2992 \fi

2993 \zref@newprop{envname}[]{\@currentvir}
2994 \zref@newprop{envline}[]{\zref@env@line}

\zref@env@line Macro \zref@env@line extracts the line number from \@currentvline.
2995 \def\zref@env@line{%
2996   \ifx\@currentvline\ltx@empty

```

```

2997 \else
2998   \expandafter
2999   \ZREF@ENV@line\@currentvline\ltx@empty line \ltx@empty\@nil
3000 \fi
3001 }

```

```
\ZREF@ENV@line
```

```
3002 \def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%
```

```
3003 </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.

7.2 Bundle installation

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

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

7.3 Package installation

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

```
tex zref.dtx
```

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

²[CTAN:pkg/zref](#)

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

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

7.4 Refresh file name databases

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

7.5 Some details for the interested

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

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

\LaTeX : Generate the documentation.

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

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

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

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

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

An example follows how to generate the documentation with pdf \LaTeX :

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

8 References

- [1] Package `footmisc`, Robin Fairbairns, 2004/01/23 v5.3a. [CTAN:pkg/footmisc](#)
- [2] Package `hyperref`, Sebastian Rahtz, Heiko Oberdiek, 2006/08/16 v6.75c. [CTAN:pkg/hyperref](#)
- [3] Package `lastpage`, Jeff Goldberg, 1994/06/25 v0.1b. [CTAN:pkg/lastpage](#)
- [4] Package `nameref`, Sebastian Rahtz, Heiko Oberdiek, 2006/02/12 v2.24. [CTAN:pkg/nameref](#)
- [5] Package `perpage`, David Kastrup, 2002/12/20 v1.0. [CTAN:pkg/perpage](#)
- [6] Package `titleref`, Donald Arseneau, 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 $\text{X}_{\text{T}}\text{T}\text{E}\text{X}$.

[2007/04/06 v1.6]

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

[2007/04/17 v1.7]

- Package atbegshi replaces everyshi.

[2007/04/22 v1.8]

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

[2007/05/02 v1.9]

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

[2007/05/06 v2.0]

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

[2007/05/28 v2.1]

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

[2008/09/21 v2.2]

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

[2008/10/01 v2.3]

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

[2009/08/07 v2.4]

- Module runs added.

[2009/12/06 v2.5]

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

[2009/12/07 v2.6]

- Version date added for package atveryend.

[2009/12/08 v2.7]

- Module titleref: Use of package gettitlestring.

[2010/03/26 v2.8]

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

[2010/03/29 v2.9]

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

[2010/04/08 v2.10]

- Module `xr` remembers the external document name in property ‘`externaldocument`’.

[2010/04/15 v2.11]

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

[2010/04/17 v2.12]

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

[2010/04/19 v2.13]

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

[2010/04/22 v2.14]

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

[2010/04/23 v2.15]

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

[2010/04/28 v2.16]

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

[2010/05/01 v2.17]

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

[2010/05/13 v2.18]

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

[2010/10/22 v2.19]

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

[2011/02/12 v2.20]

- Fix for warning in `zref-xr`.

[2011/03/18 v2.21]

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

[2011/10/05 v2.22]

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

[2011/12/05 v2.23]

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

[2012/04/04 v2.24]

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

[2016/05/16 v2.25]

- Documentation updates.

[2016/05/21 v2.26]

- update `zref-savepos` for new `luatex`

[2018/11/21 v2.27]

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

[2019/11/29 v2.28]

- Documentation updates.
- Use `iftex` directly.

[2020-03-03 v2.29]

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

[2020-03-04 v2.30]

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

[2020-05-28 v2.31]

- Adapted module zref-counter to use `\@currentcounter` in the next 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.

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> ..	<i>509, 525, 2661, 2674</i>
<code>\@PackageInfo</code>	<i>2110</i>
<code>\@PackageInfoNoLine</code>	<i>549,</i>
	<i>564, 1356, 1451, 1463, 1530, 1582</i>
<code>\@PackageWarning</code>	<i>692</i>
<code>\@addtoreset</code>	<i>1014</i>
<code>\@auxout</code>	<i>716</i>
<code>\@begintheorem</code>	<i>2049, 2054</i>
<code>\@bsphack</code>	<i>601, 611, 631, 2533</i>
<code>\@caption</code>	<i>1894</i>
<code>\@chapter</code>	<i>1906, 1943</i>
<code>\@currenthref</code>	<i>945</i>
<code>\@currentcounter</code>	<i>1036</i>
<code>\@currentlabel</code>	<i>940</i>
<code>\@currenvir</code>	<i>2993</i>
<code>\@currenvline</code>	<i>2996, 2999</i>
<code>\@ehc</code>	<i>297,</i>
	<i>307, 492, 515, 527, 1614, 2665, 2678</i>
<code>\@elt</code>	<i>1737</i>
<code>\@esphack</code>	<i>608, 628, 643, 2545</i>
<code>\@firstofone</code>	<i>1570</i>
<code>\@firstoftwo</code>	<i>1436, 1514</i>
<code>\@ifclassloaded</code>	<i>1941, 1978</i>
<code>\@ifdefinable</code>	<i>243, 290</i>
<code>\@ifl@t@r</code>	<i>223, 1034</i>
<code>\@ifnextchar</code>	<i>531, 1760</i>
<code>\@ifpackageloaded</code>	<i>1998, 2017, 2025, 2044</i>
<code>\@ifstar</code>	<i>496, 2130</i>
<code>\@ifundefined</code>	<i>192, 910,</i>
	<i>1012, 1771, 2266, 2308, 2696, 2697</i>
<code>\@input</code>	<i>2327</i>
<code>\@inputcheck</code>	<i>2158, 2159, 2174, 2211, 2213</i>
<code>\@latex@warning</code>	<i>778</i>
<code>\@mainaux</code>	<i>1699</i>
<code>\@namedef</code>	<i>536</i>
<code>\@one</code>	<i>1736</i>
<code>\@newl@bel</code>	<i>286</i>
<code>\@nil</code>	<i>1629,</i>
	<i>1631, 1859, 1866, 2048, 2049,</i>
	<i>2202, 2205, 2385, 2394, 2999, 3002</i>
<code>\@onelevel@sanitize</code>	<i>423, 441, 506, 534, 2047, 2050</i>
<code>\@opargbegintheorem</code>	<i>2039</i>
<code>\@part</code>	<i>1900</i>
<code>\@schapter</code>	<i>1924</i>

<code>\providecommand</code>	223, 282, 1692, 2077, 2950, 2951, 2952	<code>\theotype</code>	2298
<code>\ProvidesPackage</code>	<code>\thepage</code>	43, 44, 45, 714, 718, 779, 941, 1790
...	189, 221, 966, 1004, 1027, 1054, 1096, 1136, 1227, 1245, 1426, 1597, 1690, 1708, 1821, 2065, 2468, 2480, 2572, 2931, 2986	<code>\thezpage</code>	16, 1790, 1796
R		<code>\thezref@unique</code>	10, 912, 1787, 1788, 1795, 1796, 1798, 2961, 2963, 2966, 2967, 2975
<code>\read</code>	2211	<code>\title</code>	2417, 2446
<code>\ReadOnlyShipoutCounter</code>	1022	<code>\toks@</code>	424, 430, 444, 445, 554, 557, 559, 562, 613, 620, 621, 627, 741, 743, 756, 757, 1067, 1072, 1352, 1356, 1362, 1363, 1524, 1531, 1548, 1549, 1836, 1842, 2220, 2235
<code>\refstepcounter</code>	1044	<code>\topmargin</code>	1300
<code>\renewcommand</code>	7, 46, 912	<code>\TR@TitleReference</code> ..	2342, 2402, 2441
<code>\RequirePackage</code> ...	191, 196, 224, 225, 226, 227, 228, 231, 267, 272, 280, 968, 1006, 1029, 1056, 1057, 1098, 1103, 1104, 1138, 1143, 1144, 1145, 1146, 1229, 1234, 1235, 1247, 1252, 1253, 1428, 1433, 1441, 1442, 1457, 1469, 1478, 1479, 1599, 1710, 1715, 1823, 1828, 1829, 2067, 2072, 2073, 2470, 2482, 2574, 2579, 2584, 2933, 2938, 2940, 2988	<code>\trimedge</code>	1312
<code>\reset@font</code>	933	<code>\trimtop</code>	1315
<code>\rightarrow</code>	45	<code>\ttl@sect@i</code>	2000
<code>\romannumeral</code>	583, 805, 826, 1618, 2107, 2116, 2117	<code>\ttl@straight@i</code>	2006
S		U	
<code>\savepos</code>	2528	<code>\unexpanded</code>	271, 276
<code>\section</code>	63, 137, 145	<code>\UniqueCounterCall</code>	1164
<code>\setcounter</code>	919, 1018	<code>\UniqueCounterNew</code>	1147
<code>\setlength</code>	2962	<code>\uppermargin</code>	1316
<code>\SetupKeyvalOptions</code>	2081	<code>\url</code>	2437
<code>\ShipoutBoxDepth</code>	1322	<code>\urluse</code>	2432
<code>\ShipoutBoxHeight</code>	1321	<code>\usepackage</code>	9, 41, 48, 70, 72
<code>\ShipoutBoxWidth</code>	1320	V	
<code>\space</code>	779, 1452, 1464, 2179, 2180, 2184, 2188, 2193, 2490, 2502	<code>\value</code>	13, 1113, 1336, 1496
<code>\spinmargin</code>	1313	<code>\verb</code>	173
<code>\stepcounter</code>	<code>\voffset</code>	1299
...	19, 1020, 1721, 1722, 1731, 1751	W	
<code>\stockheight</code>	1272, 2697	<code>\write</code>	705, 706, 1699
<code>\stockwidth</code>	1271, 2696, 2701, 2719, 2729, 2742	X	
T		<code>\x</code>	331, 336, 724, 729, 898, 900, 1206, 1222, 1258, 1265, 1355, 1358, 1361, 1401, 1529, 1534, 1547, 1557, 1662, 1667, 1670, 1676, 1765, 1768, 2046, 2047, 2052, 2219, 2222, 2227, 2232, 2242, 2245, 2255, 2256, 2287, 2292
<code>\tableofcontents</code>	59, 132	<code>\XR@ext</code>	2078
<code>\textheight</code>	1304	Y	
<code>\textwidth</code>	1303	<code>\y</code>	2048, 2049
<code>\TeXeTstate</code>	2539	Z	
<code>\the</code>	13, 153, 155, 430, 445, 461, 557, 562, 621, 627, 743, 757, 927, 1022, 1072, 1113, 1119, 1180, 1336, 1356, 1363, 1364, 1365, 1367, 1412, 1414, 1415, 1443, 1445, 1458, 1470, 1496, 1531, 1549, 1550, 1552, 1842, 2235, 2241, 2252, 2286, 2371, 2380, 2512, 2513, 2515, 2516	<code>\z</code>	2049, 2050, 2052
<code>\thechapter</code>	14	<code>\z@</code>	2701, 2719, 2729, 2742, 2977
<code>\thefoo</code>	7, 12, 20	<code>\Z@D@page</code>	1167
		<code>\Z@L@LastPage</code>	1069
		<code>\Z@L@main</code>	1068
		<code>\Z@L@ZREF@temp</code> ..	633, 637, 640, 651, 661
		<code>\zdotfill</code>	19, 170, 173, 2954
		<code>\zdotfillsetup</code>	19, 2953
		<code>\zexternaldocument</code>	19, 2123
		<code>\ziflastpage</code>	12, 1085
		<code>\zifrefundefined</code>	8, 763

\zkvlabel 980
\zlabel 11, 83, 104, 138, 146, 973
\zlistpageattr 1491
\zlistpagelayout 15, 1331
\zmakeperpage 16, 1758
\znextpage 13, 51, 54, 1163
\znextpagesetup 14, 42, 1149
\znonextpagename 46, 1166, 1214
\zpageref 11, 126, 995
\zposx 18, 153, 2562, 2966, 2967
\zposy 18, 155, 2562
\zref 11, 25, 26, 27, 28, 112,
114, 123, 128, 129, 139, 986, 996
\ZREF@@@delprop ... 435, 437, 472, 474
\ZREF@@@newprop 539, 543
\ZREF@@delprop
..... 425, 434, 448, 465, 471, 481
\ZREF@@extract 809, 815
\ZREF@@makeperpage .. 1760, 1766, 1770
\ZREF@@newprop 517, 528, 531, 535
\ZREF@@perpage@step 1775, 1783
\ZREF@abspos@body@x@center 2881
\ZREF@abspos@body@x@left . 2879, 2906
\ZREF@abspos@body@x@right 2880, 2903
\ZREF@abspos@body@y@bottom
..... 2886, 2898, 2920
\ZREF@abspos@body@y@center 2890, 2921
\ZREF@abspos@body@y@top
..... 2882, 2887, 2891, 2919
\ZREF@abspos@foot@x@center 2896
\ZREF@abspos@foot@x@left 2894
\ZREF@abspos@foot@x@right 2895
\ZREF@abspos@foot@y@bottom 2897
\ZREF@abspos@head@x@center
..... 2861, 2881, 2896
\ZREF@abspos@head@x@left
..... 2847, 2858, 2862, 2879, 2894
\ZREF@abspos@head@x@right
..... 2857, 2880, 2895
\ZREF@abspos@head@y@bottom 2871, 2883
\ZREF@abspos@head@y@center 2875
\ZREF@abspos@head@y@top
..... 2865, 2872, 2876
\ZREF@abspos@marginpar@x@center 2915
\ZREF@abspos@marginpar@x@left ..
..... 2901, 2912, 2916
\ZREF@abspos@marginpar@x@right 2911
\ZREF@abspos@marginpar@y@bottom 2920
\ZREF@abspos@marginpar@y@center 2921
\ZREF@abspos@marginpar@y@top .. 2919
\ZREF@abspos@media@height
..... 2716, 2762, 2768
\ZREF@abspos@media@width
..... 2698, 2755, 2759
\ZREF@abspos@media@x@center ... 2757
\ZREF@abspos@media@x@left 2751, 2758
\ZREF@abspos@media@x@right 2754
\ZREF@abspos@media@y@bottom 2764
\ZREF@abspos@media@y@center ... 2767
\ZREF@abspos@media@y@top . 2761, 2779
\ZREF@abspos@origin@x ... 2788, 2849
\ZREF@abspos@origin@y ... 2818, 2867
\ZREF@abspos@paper@x@center
..... 2776, 2924
\ZREF@abspos@paper@x@left
..... 2770, 2848, 2922
\ZREF@abspos@paper@x@right 2773, 2923
\ZREF@abspos@paper@y@bottom
..... 2780, 2926
\ZREF@abspos@paper@y@center
..... 2784, 2927
\ZREF@abspos@paper@y@top
..... 2779, 2781, 2785, 2866, 2925
\ZREF@abspos@stock@x@center ... 2924
\ZREF@abspos@stock@x@left 2922
\ZREF@abspos@stock@x@right 2923
\ZREF@abspos@stock@y@bottom ... 2926
\ZREF@abspos@stock@y@center ... 2927
\ZREF@abspos@stock@y@top 2925
\ZREF@abspos@used 2648, 2650
\ZREF@absposnum 2607, 2619, 2625
\zref@absposnumused 2670
\zref@absposnumx ... 2589, 2601, 2639
\zref@absposnumy ... 2596, 2613, 2642
\zref@absposused 2647
\zref@absposx 2587, 2633
\zref@absposy 2594, 2636
\zref@addprop
. 6, 76, 360, 1023, 1037, 1041,
1106, 1109, 1263, 1279, 1283,
1287, 1291, 1444, 1446, 1459,
1471, 1646, 1831, 2476, 2585, 2586
\zref@addprops
. 6, 15, 341, 942, 1329, 1718, 2518
\ZREF@addtoks 755
\ZREF@base@ok 962, 969, 1007,
1030, 1058, 1099, 1139, 1230,
1248, 1429, 1600, 1711, 1824,
2068, 2471, 2483, 2575, 2934, 2989
\ZREF@call 1172, 1187, 1196, 1200, 1208
\ZREF@def@abspos
..... 2633, 2636, 2639, 2642, 2644
\zref@def@absposnumx 2638
\ZREF@def@absposnumy 2644
\zref@def@absposnumy 2641
\zref@def@absposx 2632
\zref@def@absposy 2635
\ZREF@def@extract 846, 848
\zref@def@extract 8, 845
\ZREF@def@extractdefault ... 857, 859
\zref@def@extractdefault 856
\ZREF@default 562, 563, 572
\zref@default ... 8, 531, 807, 930, 932
\ZREF@delprop
..... 413, 416, 418, 453, 456, 458
\zref@delprop 412, 452
\ZREF@df@dot 2948, 2952, 2981
\ZREF@df@min 2945, 2950, 2968
\ZREF@df@unit 2942, 2951, 2962
\ZREF@dotfill 2964, 2971, 2980
\ZREF@ENV@line 2999, 3002
\zref@env@line 2994, 2995
\ZREF@extract 804, 821, 824, 876

<code>\zref@extract</code>	7, 95, 96, 109, 140, 804, 824, 853, 871, 876, 993, 1122, 1218, 1367, 1414, 1415, 1538, 1575, 1795, 1796, 1889, 2563, 2566, 2755, 2759, 2762, 2768, 2774, 2777, 2782, 2786, 2807, 2836, 2850, 2852, 2854, 2859, 2863, 2868, 2869, 2873, 2877, 2884, 2888, 2892, 2899, 2904, 2907, 2908, 2913, 2917	<code>\zref@localaddprops</code>	376
<code>\ZREF@extractdefault</code> 825, 841, 844, 875		<code>\zref@localdelprop</code>	415, 455, 669
<code>\zref@extractdefault</code>	7, 116, 117, 817, 844, 864, 870, 875, 1078, 1079, 1176, 1191, 1237, 1798, 2340, 2343, 2344, 2348, 2349, 2352, 2354, 2355, 2357, 2359, 2590, 2597, 2656, 2684, 2851, 2902	<code>\ZREF@mainlist</code>	598, 936, 939, 942, 1023, 1037, 1041, 1831, 2476
<code>\ZREF@false</code>	677, 687	<code>\ZREF@makeperpage@opt</code>	1760, 1763
<code>\ZREF@foundfalse</code>	2401	<code>\ZREF@MARKS@DefineProp</code>	1622, 1623, 1624, 1658
<code>\ZREF@foundtrue</code>	2448	<code>\zref@marks@register</code>	1604, 1609, 1641, 1680
<code>\ZREF@getcurrent</code>	582, 593, 596, 874	<code>\ZREF@name</code>	229, 259, 270, 292, 295, 305, 346, 364, 381, 399, 490, 509, 521, 525, 549, 564, 616, 680, 692, 1608, 2489, 2501
<code>\zref@getcurrent</code>	7, 596, 869, 874	<code>\ZREF@NAME@bot</code>	1637, 1657
<code>\zref@hex</code>	1443, 1445, 1458, 1470, 1484, 1487	<code>\ZREF@NAME@first</code>	1636, 1656
<code>\zref@ifabsposnumundefined</code> 2683, 2688		<code>\ZREF@NAME@top</code>	1635, 1655
<code>\zref@ifabsposundefined</code>	2681	<code>\zref@newlabel</code>	7, 282, 285, 750, 2239, 2325
<code>\ZREF@ifDefinable</code>	242, 763, 995, 998, 1085, 1127, 1163, 1331, 1491, 1758, 1812, 1879, 1882, 2123, 2547, 2552, 2557, 2954	<code>\zref@newlist</code>	6, 288, 939, 1062, 1105, 1632, 1717, 2510
<code>\ZREF@iflastpage</code>	1086, 1088, 1088	<code>\ZREF@newprop</code>	498, 501, 504
<code>\zref@iflastpage</code>	12, 1077, 1091	<code>\zref@newprop</code>	6, 12, 13, 14, 75, 495, 940, 941, 944, 951, 955, 959, 1022, 1036, 1040, 1260, 1276, 1280, 1284, 1288, 1320, 1321, 1322, 1325, 1326, 1327, 1443, 1445, 1458, 1470, 1663, 1671, 2074, 2075, 2076, 2396, 2512, 2513, 2515, 2516, 2993, 2994
<code>\zref@iflistcontainsprop</code>	6, 310, 345, 363, 380, 398, 648, 658	<code>\ZREF@NewPropAnchor</code>	943, 2125, 2475
<code>\zref@iflistundefined</code>	6, 289, 300, 304, 311	<code>\ZREF@NewPropPageValue</code> 958, 1107, 1716	
<code>\zref@ifpropundefined</code> 7, 485, 489, 519, 548, 615, 828, 1410, 1659, 2395		<code>\ZREF@NewPropTheotype</code>	954, 2296
<code>\ZREF@ifrefcontainsprop</code>	787, 795	<code>\ZREF@NewPropTitle</code>	949, 1830, 2126
<code>\zref@ifrefcontainsprop</code>	8, 783, 1412, 2356, 2452, 2453	<code>\ZREF@nextpage</code>	1164, 1168
<code>\ZREF@ifrefundefined</code>	765, 768, 1173, 1184, 1194	<code>\ZREF@nil</code> 545, 797, 836, 2212, 2218, 2224, 2229, 2239, 2255, 2284, 2292, 2383, 2390, 2399, 2402, 2441	
<code>\zref@ifrefundefined</code>	8, 760, 770, 776, 784, 827, 1185, 1365, 1552, 1788, 2604, 2616, 2652, 2682, 2690, 2963	<code>\ZREF@NOVALUE</code>	803
<code>\ZREF@immediatefalse</code>	678	<code>\ZREF@novalue</code>	796, 797, 803
<code>\ZREF@immediatetrue</code>	675, 704	<code>\ZREF@np@call@next</code>	1158, 1162, 1217
<code>\ZREF@label</code> 603, 627, 637, 640, 710, 1072		<code>\ZREF@np@call@nonext</code> 1155, 1161, 1213	
<code>\zref@label</code>	7, 597, 977	<code>\ZREF@np@call@unknown</code> 1151, 1160, 1209	
<code>\zref@labelbykv</code>	630, 984	<code>\ZREF@np@setup@i</code>	1150, 1153
<code>\zref@labelbylist</code>	7, 598, 600, 1113, 1787, 2549	<code>\ZREF@np@setup@ii</code>	1154, 1157
<code>\zref@labelbyprops</code>	7, 88, 610, 1171, 2554, 2559, 2961, 2975	<code>\ZREF@number</code> 924, 1538, 1544, 1606, 2689	
<code>\zref@listexists</code>	6, 303, 322, 342, 361, 377, 396, 419, 459, 602	<code>\ZREF@org@@@begintheorem</code>	2056
<code>\zref@listforloop</code>	321, 657	<code>\ZREF@org@@@caption</code>	1896
<code>\zref@listpageattr</code>	1491	<code>\ZREF@org@@@chapter</code>	1908, 1964
<code>\zref@listpagelayout</code>	1331	<code>\ZREF@org@@@opargbegintheorem</code>	2041
<code>\zref@localaddprop</code>	395	<code>\ZREF@org@@@part</code>	1902
		<code>\ZREF@org@@@schapter</code>	1926
		<code>\ZREF@org@@@sect</code>	1914
		<code>\ZREF@org@@@spart</code>	1920
		<code>\ZREF@org@@@ssect</code>	1932
		<code>\ZREF@org@@@stpelt</code>	1747, 1752, 1756
		<code>\ZREF@org@beamer@section</code>	1982
		<code>\ZREF@org@beamer@subsection</code>	1988
		<code>\ZREF@org@beamer@subsubsection</code> 1994	
		<code>\ZREF@org@descriptionlabel</code>	1938
		<code>\ZREF@org@lst@MakeCaption</code>	2034
		<code>\ZREF@org@LT@c@ption</code>	2020
		<code>\ZREF@org@M@sect</code>	1973

<code>\ZREF@org@refstepcounter</code>	1046	<code>\zref@require@unique</code>	
<code>\ZREF@org@stepcounter</code> 1721, 1726, 1751		10, 909, 1720, 2939
<code>\ZREF@org@testdef</code>		<code>\ZREF@Robust</code>	232,
.	1338, 1340, 1498, 1500	238, 244, 285, 288, 303, 310,	
<code>\ZREF@org@thepage</code>	714, 718	341, 360, 376, 395, 412, 415,	
<code>\ZREF@org@ttl@sect@i</code>	2002	452, 455, 488, 495, 547, 577,	
<code>\ZREF@org@ttl@straight@i</code>	2013	597, 600, 610, 630, 702, 772,	
<code>\ZREF@org@write</code>	705, 706	845, 856, 867, 883, 909, 929,	
<code>\ZREF@P</code>	505,	935, 1124, 1543, 1578, 1834,	
506, 508, 510, 519, 522, 526,		1847, 2632, 2635, 2638, 2641, 2647	
536, 537, 539, 540, 541, 545,		<code>\ZREF@SavedEscapechar</code>	461, 468
721, 725, 726, 735, 739, 744, 745		<code>\zref@savepos</code>	18, 2519, 2535, 2541
<code>\ZREF@pa@AfterLastShipout</code> 1494, 1592		<code>\ZREF@savepos@ok</code>	2568, 2580
<code>\ZREF@pa@AtVeryEnd</code>	1507, 1510, 1581	<code>\zref@setcurrent</code>	
<code>\ZREF@pa@ListPage</code>	1528, 1546	6, 81, 541, 577, 697, 1045
<code>\ZREF@pa@listtrue</code>	1492	<code>\zref@setdefault</code>	8, 929, 932
<code>\ZREF@page@max</code>	1336, 1402, 1496, 1558	<code>\zref@setmainlist</code>	9, 935
<code>\zref@pageattr</code>	1536	<code>\zref@showprop</code>	547
<code>\zref@pageattr@used</code>	1543	<code>\ZREF@STAR</code>	1630, 1654
<code>\ZREF@pagenum@last</code>	1190, 1193	<code>\ZREF@stripperperiod</code>	1858, 1866
<code>\ZREF@pagenum@this</code>		<code>\ZREF@temp</code>	193, 200, 201, 202,
.	1175, 1180, 1183, 1193, 1199	203, 204, 205, 206, 207, 208,	
<code>\ZREF@par</code>	508, 533	209, 210, 211, 212, 213, 214,	
<code>\ZREF@param</code>		215, 216, 233, 234, 440, 441,	
.	422, 423, 442, 460, 477, 646,	442, 720, 741, 742, 750, 1254,	
647, 648, 652, 673, 674, 677, 682		1268, 1269, 1270, 1271, 1272,	
<code>\ZREF@patch</code>	249, 1043, 1893, 1899,	1273, 1274, 1293, 1294, 1295,	
1905, 1911, 1917, 1923, 1929,		1296, 1298, 1299, 1300, 1301,	
1935, 1966, 1979, 1985, 1991,		1302, 1303, 1304, 1305, 1306,	
1999, 2005, 2018, 2026, 2038, 2053		1307, 1308, 1309, 1310, 1311,	
<code>\zref@pdfpageattr</code>		1312, 1313, 1314, 1315, 1316,	
.	1454, 1536, 1542, 1553	1317, 1341, 1342, 1434, 1447,	
<code>\zref@pdfpageattr@used</code>	1455	1460, 1472, 1475, 1480, 1481,	
<code>\zref@pdfpagesattr</code>	1466, 1573, 1586	1482, 1483, 1501, 1502, 1510,	
<code>\zref@pdfpagesattr@used</code>	1467, 1578	1511, 1512, 1513, 1566, 1567,	
<code>\ZREF@pl@AfterLastShipout</code> 1334, 1421		1568, 1569, 1629, 1630, 2007,	
<code>\ZREF@pl@AtVeryEnd</code>	1347, 1350	2008, 2099, 2109, 2112, 2116,	
<code>\ZREF@pl@ListEntry</code>		2655, 2658, 2659, 2788, 2791,	
.	1369, 1370, 1371, 1372, 1373,	2796, 2800, 2802, 2805, 2818,	
1374, 1375, 1376, 1377, 1378,		2821, 2826, 2829, 2831, 2834, 2959	
1379, 1380, 1381, 1382, 1383,		<code>\ZREF@TempName</code>	1605, 1617, 1618,
1384, 1385, 1386, 1387, 1388,		1620, 1646, 1659, 1663, 1671, 1682	
1389, 1390, 1391, 1392, 1393,		<code>\ZREF@TempNum</code>	
1394, 1395, 1396, 1397, 1398, 1409		1606, 1607, 1611, 1618, 1660, 1673	
<code>\ZREF@pl@ListPage</code>	1354, 1360	<code>\zref@thepage</code>	13, 1121, 1130
<code>\ZREF@pl@listtrue</code>	1332	<code>\zref@thepage@atbegshi@hook</code>	
<code>\zref@pos@label@used</code>	2662	1108, 1112
<code>\zref@pos@num@used</code>	2676	<code>\zref@thepage@name</code>	
<code>\zref@prop</code> 324, 332, 333, 337, 658, 662		13, 1116, 1122, 1125, 1179
<code>\zref@propexists</code>	7, 344, 362,	<code>\zref@thepage@refused</code>	1124, 1129
379, 397, 488, 578, 647, 668, 987		<code>\ZREF@titleref</code>	1883, 1885
<code>\ZREF@refname@next</code>		<code>\zref@titleref@cleanup</code>	1834, 1874
.	1178, 1185, 1194, 1218	<code>\zref@titleref@current</code>	
<code>\ZREF@refname@this</code>	949, 1853, 1857, 1858, 1877
.	1170, 1171, 1173, 1176	<code>\ZREF@titleref@hook</code>	
<code>\ZREF@RefPrefix</code>	284, 286, 1342, 1502	1833, 1837, 1841, 1864
<code>\ZREF@refused</code>	773, 775	<code>\zref@titleref@setcurrent</code>	
<code>\zref@refused</code>	1847, 1895, 1901, 1907, 1913,
.	8, 769, 772, 849, 860, 992,	1919, 1925, 1931, 1937, 1945,	
999, 1089, 1090, 1125, 1240,		1948, 1952, 1956, 1958, 1969,	
1544, 1579, 1887, 2651, 2659, 2672		1971, 1981, 1987, 1993, 2001,	
		2009, 2011, 2021, 2030, 2040, 2055	

<code>\zref@titleref@stripperperiodtrue</code>	1846	<code>\ZREF@xr@list</code>	2245, 2246
<code>\ZREF@true</code>	674, 688	<code>\ZREF@xr@ltx@ignorewarning</code>	2374
<code>\ZREF@u@getcurrent</code>	591	<code>\ZREF@xr@newlabel</code>	2227, 2326
<code>\zref@unhex</code>	1485, 1488, 1537, 1574	<code>\ZREF@xr@prefix</code>	2140, 2240, 2276, 2280, 2285, 2311, 2318, 2322
<code>\ZREF@UpdatePdfTeX</code>	247, 2492, 2505	<code>\ZREF@xr@process@label</code>	2229, 2284
<code>\ZREF@value</code>	557, 558, 571	<code>\ZREF@xr@process@zreflabel</code>	2224, 2239
<code>\ZREF@wrapper@babel</code>	900, 906	<code>\ZREF@xr@processfile</code>	2157, 2210
<code>\zref@wrapper@babel</code>	10, 140, 765, 773, 846, 857, 883, 977, 984, 988, 1086, 1883, 2633, 2636, 2639, 2642, 2648	<code>\ZREF@xr@processline</code>	2212, 2218
<code>\zref@wrapper@immediate</code>	10, 87, 636, 702, 1071	<code>\ZREF@xr@refname</code>	2240, 2266, 2273, 2285, 2308, 2315
<code>\ZREF@wrapper@unexpanded</code>	867, 881	<code>\ZREF@xr@relax</code>	2328, 2415
<code>\zref@wrapper@unexpanded</code>	10, 868, 873, 878, 2335	<code>\ZREF@xr@scanparams</code>	2290, 2399
<code>\ZREF@wu@extract</code>	819, 871	<code>\ZREF@xr@scantitleref</code>	2402, 2441
<code>\ZREF@wu@extractdefault</code>	839, 870	<code>\ZREF@xr@temp</code>	2414, 2415
<code>\ZREF@wu@getcurrent</code>	591, 869	<code>\ZREF@xr@tempname</code>	2243, 2244, 2264, 2269, 2280, 2288, 2289, 2306, 2322
<code>\ZREF@X</code>	497, 500, 537	<code>\ZREF@xr@temprefname</code>	2244, 2256, 2258, 2274, 2289, 2291, 2298, 2301, 2316
<code>\zref@xr@</code>	2089	<code>\ZREF@xr@theURL</code>	2107, 2109, 2111, 2117, 2152, 2433
<code>\ZREF@xr@AddUrl</code>	2101, 2104	<code>\ZREF@xr@tolabel</code>	2280, 2322, 2329
<code>\ZREF@xr@input</code>	2232, 2327	<code>\ZREF@xr@URL</code>	2095, 2105, 2106, 2107
<code>\ZREF@xr@AddURL</code>	2097, 2150, 2429	<code>\ZREF@xr@url</code>	2148, 2150, 2151, 2459
<code>\ZREF@xr@checkfile</code>	2154, 2157, 2207	<code>\ZREF@xr@urlcheck</code>	2264, 2306, 2451
<code>\ZREF@xr@checkkey</code>	2385, 2394	<code>\ZREF@xr@zref@ignorewarning</code>	2276, 2318, 2365
<code>\ZREF@xr@checklist</code>	2255, 2383	<code>\ZREF@xr@zref@newlabel</code>	2222, 2325
<code>\zref@xr@ext</code>	20, 2077, 2145	<code>\ZREF@xr@zreflabelfalse</code>	2131
<code>\ZREF@xr@externaldocument</code>	2132, 2135, 2139	<code>\ZREF@xr@zreflabeltrue</code>	2134
<code>\ZREF@xr@externalfile</code>	2142, 2143, 2261, 2303	<code>\ZREF@zref</code>	988, 991
<code>\ZREF@xr@file</code>	2143, 2158, 2161, 2167, 2178, 2201, 2249, 2368, 2377	<code>\ZREF@zsavepos</code>	2532, 2549, 2554, 2559
<code>\ZREF@xr@filelist</code>	2141, 2199, 2202, 2204, 2205, 2233, 2234	<code>\zrefused</code>	11, 92, 93, 161, 162, 163, 998
<code>\ZREF@xr@found</code>	2169, 2179, 2241, 2286	<code>\zruns</code>	16, 1692
<code>\ZREF@xr@graburl</code>	2145, 2147	<code>\zsavepos</code>	17, 157, 158, 2547
<code>\ZREF@xr@ignored@empty</code>	2170, 2182, 2184, 2251, 2252	<code>\zsaveposx</code>	17, 2552
<code>\ZREF@xr@ignored@ltx</code>	2172, 2191, 2193, 2379, 2380	<code>\zsaveposy</code>	2557
<code>\ZREF@xr@ignored@zref</code>	2171, 2186, 2188, 2370, 2371	<code>\zthepage</code>	13, 1127
<code>\ZREF@xr@line</code>	2211, 2212, 2224, 2229	<code>\ztitleref</code>	17, 1882
		<code>\ztitlerefsetup</code>	17, 1867
		<code>\ztotpages</code>	15, 124, 1236
		<code>\zunknownnextpagename</code>	14, 1167, 1210
		<code>\zumakeperpage</code>	16, 1812
		<code>\zxrsetup</code>	19, 2092