

The zref package

Heiko Oberdiek*

2020-07-03 v2.32

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

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

| | | |
|----------|--|-----------|
| 3.11 | Module counter | 18 |
| 3.12 | Module titleref | 18 |
| 3.13 | Module savepos | 19 |
| 3.14 | Module abspos | 20 |
| 3.15 | Module dotfill | 20 |
| 3.16 | Module env | 21 |
| 3.17 | Module xr | 21 |
| 3.18 | Module pageattr | 22 |
| 4 | ToDo | 22 |
| 5 | Example | 22 |
| 6 | Implementation | 25 |
| 6.1 | Package zref | 25 |
| 6.1.1 | Identification | 25 |
| 6.1.2 | Load basic module | 25 |
| 6.1.3 | Process options | 25 |
| 6.2 | Module base | 25 |
| 6.2.1 | Prefixes | 25 |
| 6.2.2 | Identification | 26 |
| 6.2.3 | Utilities | 26 |
| 6.2.4 | Check for ϵ -TeX | 27 |
| 6.2.5 | Auxiliary file stuff | 27 |
| 6.2.6 | Property lists | 28 |
| 6.2.7 | Properties | 32 |
| 6.2.8 | Reference generation | 35 |
| 6.2.9 | Reference querying and extracting | 38 |
| 6.2.10 | Compatibility with babel | 41 |
| 6.2.11 | Unique counter support | 42 |
| 6.2.12 | Utilities | 42 |
| 6.2.13 | Setup | 42 |
| 6.3 | Module user | 44 |
| 6.4 | Module abspage | 45 |
| 6.5 | Module counter | 45 |
| 6.6 | Module lastpage | 46 |
| 6.7 | Module thepage | 47 |
| 6.8 | Module nextpage | 48 |
| 6.9 | Module totpages | 49 |
| 6.10 | Module pagelayout | 50 |
| 6.10.1 | Define layout properties | 50 |
| 6.11 | Module pageattr | 53 |
| 6.12 | Module marks | 57 |
| 6.13 | Module runs | 58 |
| 6.14 | Module perpage | 59 |
| 6.15 | Module titleref | 61 |
| 6.15.1 | Implementation | 61 |
| 6.15.2 | User interface | 63 |
| 6.15.3 | Patches for section and caption commands | 63 |
| 6.15.4 | Environment description | 64 |
| 6.15.5 | Class memoir | 64 |
| 6.15.6 | Class beamer | 65 |
| 6.15.7 | Package titlesec | 65 |
| 6.15.8 | Package longtable | 66 |

| | | |
|--------------------|---------------------------------|-----------|
| 6.15.9 | Package listings | 66 |
| 6.15.10 | Theorems | 66 |
| 6.16 | Module <code>xr</code> | 67 |
| 6.17 | Module <code>hyperref</code> | 75 |
| 6.18 | Module <code>savepos</code> | 75 |
| 6.18.1 | Identification | 75 |
| 6.18.2 | Availability | 75 |
| 6.18.3 | Setup | 76 |
| 6.18.4 | User macros | 76 |
| 6.19 | Module <code>abspos</code> | 77 |
| 6.19.1 | Identification | 77 |
| 6.19.2 | Media | 80 |
| 6.19.3 | Paper | 82 |
| 6.19.4 | Origin | 82 |
| 6.19.5 | Header | 83 |
| 6.19.6 | Body | 84 |
| 6.19.7 | Footer | 85 |
| 6.19.8 | Marginal notes | 85 |
| 6.19.9 | Stock paper | 86 |
| 6.20 | Module <code>dotfill</code> | 86 |
| 6.21 | Module <code>env</code> | 87 |
| 7 | Installation | 88 |
| 7.1 | Download | 88 |
| 7.2 | Bundle installation | 88 |
| 7.3 | Package installation | 88 |
| 7.4 | Refresh file name databases | 89 |
| 7.5 | Some details for the interested | 89 |
| 8 | References | 90 |
| 9 | History | 90 |
| [2006/02/20 v1.0] | | 90 |
| [2006/05/03 v1.1] | | 90 |
| [2006/05/25 v1.2] | | 90 |
| [2006/09/08 v1.3] | | 90 |
| [2007/01/23 v1.4] | | 90 |
| [2007/02/18 v1.5] | | 91 |
| [2007/04/06 v1.6] | | 91 |
| [2007/04/17 v1.7] | | 91 |
| [2007/04/22 v1.8] | | 91 |
| [2007/05/02 v1.9] | | 91 |
| [2007/05/06 v2.0] | | 91 |
| [2007/05/28 v2.1] | | 91 |
| [2008/09/21 v2.2] | | 91 |
| [2008/10/01 v2.3] | | 91 |
| [2009/08/07 v2.4] | | 91 |
| [2009/12/06 v2.5] | | 92 |
| [2009/12/07 v2.6] | | 92 |
| [2009/12/08 v2.7] | | 92 |
| [2010/03/26 v2.8] | | 92 |
| [2010/03/29 v2.9] | | 92 |
| [2010/04/08 v2.10] | | 92 |
| [2010/04/15 v2.11] | | 92 |

| | |
|--------------------|----|
| [2010/04/17 v2.12] | 92 |
| [2010/04/19 v2.13] | 93 |
| [2010/04/22 v2.14] | 93 |
| [2010/04/23 v2.15] | 93 |
| [2010/04/28 v2.16] | 93 |
| [2010/05/01 v2.17] | 93 |
| [2010/05/13 v2.18] | 94 |
| [2010/10/22 v2.19] | 94 |
| [2011/02/12 v2.20] | 94 |
| [2011/03/18 v2.21] | 94 |
| [2011/10/05 v2.22] | 94 |
| [2011/12/05 v2.23] | 94 |
| [2012/04/04 v2.24] | 94 |
| [2016/05/16 v2.25] | 94 |
| [2016/05/21 v2.26] | 94 |
| [2018/11/21 v2.27] | 95 |
| [2019/11/29 v2.28] | 95 |
| [2020-03-03 v2.29] | 95 |
| [2020-03-04 v2.30] | 95 |
| [2020-05-28 v2.31] | 95 |
| [2020-07-03 v2.32] | 95 |

10 Index **95**

1 Introduction

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

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

1.1 Standard L^AT_EX behaviour

References are created by the `\label` command:

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

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

```
\r@myref → {2.1}{7}
```

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

There are several tries to add new properties:

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

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

1.2 Basic idea

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

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

The entries have the following format:

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

This approach is much more flexible:

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

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

1.3 Interfaces

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

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

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

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

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

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

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

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

2 Interface for programmers

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

2.1 Entities

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

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

The generic format of an entry is:

```
\Z@R@<refname> → \<propname>{<value>}
```

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

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

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

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

2.2 Property list

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

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

Declares a new empty property list.

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

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

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

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

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

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

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

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

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

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

2.3 Property

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

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

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

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

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

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

`\zref@getcurrentexp2 {<propname>}`

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

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

`\zref@propexists {<propname>} {<then>}`

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

`\zref@ifpropundefinedexp {<propname>} {<then>} {<else>}`

Calls *<then>* or *<else>* depending on the existence of property *<propname>*.

2.4 Reference generation

`\zref@label {<refname>}`

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

`\zref@labelbylist` $\{\langle refname \rangle\} \{\langle listname \rangle\}$

Same as `\zref@label` except that the properties are taken from the specified property list $\langle listname \rangle$.

`\zref@labelbyprops` $\{\langle refname \rangle\} \{\langle propnameA \rangle, \langle propnameB \rangle, \dots\}$

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

`\zref@newlabel` $\{\langle refname \rangle\} \{ \dots \}$

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

2.5 Data extraction

`\zref@extractdefault`^{exp2} $\{\langle refname \rangle\} \{\langle propname \rangle\} \{\langle default \rangle\}$

This is the basic command that references the value of a property $\langle propname \rangle$ for the reference $\langle refname \rangle$. In case of errors such as undefined reference the $\langle default \rangle$ is used instead.

`\zref@extract`^{exp2} $\{\langle refname \rangle\} \{\langle propname \rangle\}$

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` $\{\langle refname \rangle\}$

This command is not expandable. It causes the warnings if the reference $\langle refname \rangle$ 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` $\{\langle cmd \rangle\} \{\langle refname \rangle\} \{\langle propname \rangle\}$
`\zref@def@extractdefault` $\{\langle cmd \rangle\} \{\langle refname \rangle\} \{\langle propname \rangle\} \{\langle default \rangle\}$

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

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

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

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

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

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

Test whether a reference provides a property.

2.6 Setup

`\zref@default`

Holds the global default for unknown values.

`\zref@setdefault {<value>}`

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

`\zref@setmainlist {<value>}`

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

2.7 Declared properties

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

¹Module pagelayout only defines properties if the parameter exists.

2.8 Wrapper for advanced situations

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

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

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

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

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

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

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

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

2.9 Counter for unique names

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

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

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

```
\thezref@unique
```

This command is used to generate unique label names.

3 User interface

3.1 Module user

The user interface for this package and its modules is enabled by `zref's` package option `user` or package `zref-user`. The names of user commands are prefixed by `z` in

order to avoid name clashes with existing macros of the same functionality. Thus the package does not disturb the traditional reference scheme, both can be used together.

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

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

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

`\zlabel {⟨refname⟩}^babel`

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

`\zref [⟨propname⟩] {⟨refname⟩}^babel`

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

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

`\zpageref {⟨refname⟩}^babel`

Convenience macro, similar to `\pageref`.

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

`\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`

With the help of package `atbegshi` a new counter `abspage` with absolute page numbers is provided. For technical and historical reasons the counter itself 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` the first page will be page 1 as expected. Also a new property `abspage` is defined and added to the main property list. Thus you can reference the absolute page number:

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

The example also makes use of module `lastpage`.

3.3 Module `lastpage`

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

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

or

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

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

3.3.1 Tests for last page

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

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

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

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

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

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

3.3.2 Example

```
1 <*example-lastpage>
2 %<<END_EXAMPLE
3 \NeedsTeXFormat{LaTeX2e}
4 \documentclass{report}
5
6 \newcounter{foo}
7 \renewcommand*{\thefoo}{\Alph{foo}}
8
9 \usepackage{zref-lastpage,zref-user}[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 \end{example-lastpage}

```

3.4 Module `thepage`

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

`\zthepage {absolute page number}`

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

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

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

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

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

3.5 Module `nextpage`

`\znextpage`

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

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

3.5.1 Configuration

The behaviour can be configured by the following macros.

`\zunknownnextpagename`
`\znonextpagename`

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

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

`\znextpagesetup` `{\unknown}` `{\no next}` `{\next}`

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

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

3.5.2 Example

```
37 (*example-nextpage)
38 %<<END_EXAMPLE
39 \documentclass{book}
40
41 \usepackage{zref-nextpage}[2019/11/29]
42 \znextpagesetup
43   {\thepage}% next page is unknown
44   {\thepage\ (#1)}% this page is last page
45   {\thepage\ $\rightarrow$ #1}% next page is known
46 \renewcommand*{\znonextpagename}{last page}
47
48 \usepackage{fancyhdr}
```

```

49 \pagestyle{fancy}
50 \fancyhf{}
51 \fancyhead[LE,RO]{\znextpage}
52 \fancypagestyle{plain}{%
53   \fancyhf{}%
54   \fancyhead[LE,RO]{\znextpage}%
55 }
56
57 \begin{document}
58 \frontmatter
59   \tableofcontents
60 \mainmatter
61   \chapter{Hello World}
62   \clearpage
63   \section{Last section}
64 \end{document}
65 %END_EXAMPLE
66 \end{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> | mag | |
| <code>\paperwidth</code> | paperwidth | |
| <code>\paperheight</code> | paperheight | |
| <code>\stockwidth</code> | stockwidth | class memoir |
| <code>\stockheight</code> | stockheight | class memoir |
| <code>\pdfpagewidth</code> | pdfpagewidth | pdfTEX, LuaTEX |
| <code>\pdfpageheight</code> | pdfpageheight | pdfTEX, LuaTEX |
| <code>\pdfhorigin</code> | pdfhorigin | pdfTEX, LuaTEX |
| <code>\pdfvorigin</code> | pdfvorigin | pdfTEX, LuaTEX |
| <code>\hoffset</code> | hoffset | |
| <code>\voffset</code> | voffset | |
| <code>\topmargin</code> | topmargin | |
| <code>\oddsidemargin</code> | oddsidemargin | |
| <code>\evensidemargin</code> | evensidemargin | |
| <code>\textwidth</code> | textwidth | |
| <code>\textheight</code> | textheight | |
| <code>\headheight</code> | headheight | |
| <code>\headsep</code> | headsep | |
| <code>\footskip</code> | footskip | |
| <code>\marginparwidth</code> | marginparwidth | |
| <code>\marginparsep</code> | marginparsep | |
| <code>\columnwidth</code> | columnwidth | |
| <code>\columnsep</code> | columnsep | |

`\zlistpagelayout`

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

3.8 Module marks

ToDo.

3.9 Module runs

Module `runs` counts the L^AT_EX runs since last `.aux` file creation and prints the number in the `.log` file.

`\zrunsexp`

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

3.10 Module perpage

With `\@addtoreset` or `\numberwithin` a counter can be reset if another counter is incremented. This do not work well if the other counter is the page counter. The page counter is incremented in the output routine that is often called asynchronous

somewhere on the next page. A reference mechanism costs at least two L^AT_EX runs, but ensures correct page counter values.

`\zmakeperpage [reset] {counter}`

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

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

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

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

`\thezpage`
counter `zpage`

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

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

`\zunmakeperpage {counter}`

The reset mechanism for this counter is deactivated.

3.11 Module `counter`

This option just add the property `counter` to the main property list. The property stores the counter name, that was responsible for the reference. This is the property `hyperref`'s `\autoref` feature uses. Thus this property `counter` may be useful for a 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 XeTeX). pdfTeX is able to tell the current position on the page. The page position is not instantly known. First the page must be constructed by TeX's asynchronous output routine. Thus the time where the position is known is the page shipout time. Thus a reference system where the information is recorded in the first run and made available for use in the second run comes in handy.

`\zsavepos {<refname>}`

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

`\zsaveposx {<refname>}`

`\zsaveposy {<refname>}`

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

`\zposxexp {<refname>}`

`\zposyexp {<refname>}`

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

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

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

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

`\zref@savepos`

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

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

3.14 Module `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`= \langle *count value* \rangle
If the actual number of dots are smaller than \langle *count value* \rangle , then the dots are suppressed. Default: 2.

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

`dot`= \langle *value* \rangle
The dot itself is given by \langle *value* \rangle . 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 * [\langle prefix \rangle] babel { \langle external document \rangle } [\langle url \rangle]`

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

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

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

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

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

The following setup options are available:

ext: It sets the default extension.

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

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

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

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

`\zref@xr@ext`

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

3.18 Module `pageattr`

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

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

4 ToDo

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

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

5 Example

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

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

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

```
74 \makeatletter  
75 \zref@newprop{chaptitle}{}  
76 \zref@addprop{main}{chaptitle}  
77  
78 \newcommand*\ChapterStart}[2]{%  
79   \cleardoublepage  
80   \def\current@chapid{#1}%  
81   \zref@setcurrent{chaptitle}{#2}%  
82   \chapter{#2}%  
83   \zlabel{chap:#1}%  
84 }  
85 \newcommand*\ChapterStop}{%
```

```

86 \cleardoublepage
87 \zref@wrapper@immediate{%
88   \zref@labelbyprops{chapend:\current@chapid}{abspage}%
89 }%
90 }

```

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

```

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

```

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

```

101 \makeatletter
102
103 \frontmatter
104 \zlabel{documentstart}
105
106 \begin{itemize}
107 \item
108   The frontmatter part has
109   \number\numexpr\zref@extract{chap:first}{abspage}-1\relax
110   ~pages.
111 \item
112   Chapter \zref{chap:first} has \ChapterPages{first} page(s).
113 \item
114   Section \zref{hello} is on the
115   \ifcase\numexpr
116     \zref@extractdefault{hello}{page}{0}%
117     -\zref@extractdefault{chap:first}{page}{0}%
118     +1\relax
119     ??\or first\or second\or third\or forth\fi
120   ~page inside its chapter.
121 \item
122   The document has
123   \zref[abspage]{LastPage} pages.
124   This number is \ifodd\ztotpages odd\else even\fi.
125 \item
126   The last page is labeled with \zpageref{LastPage}.
127 \item
128   The title of chapter \zref{chap:next} %
129   is ‘‘\zref[chaptitle]{chap:next}’’.
130 \end{itemize}
131
132 \tableofcontents
133
134 \mainmatter
135 \ChapterStart{first}{First chapter}
136

```

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

examples with `\numexpr`.

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

Here an example follows that makes use of pdfTeX's "savepos" feature. The position on the page is not known before the page is constructed and shipped out. Therefore the position is stored in references and are available for calculations in the next L^AT_EX compile run.

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

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

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

Test for module `\dotfill`.

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



```
186 </example>
```

6 Implementation

6.1 Package zref

6.1.1 Identification

```
187 <*package>
188 \NeedsTeXFormat{LaTeX2e}
189 \ProvidesPackage{zref}
190 [2020-07-03 v2.32 A new reference scheme for LaTeX (HO)]%
```

6.1.2 Load basic module

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

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

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

6.1.3 Process options

Known modules are loaded and the release date is checked.

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

217 \ProcessOptions\relax
218 </package>
```

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:

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

\ZREF@: Internal macros.

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

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

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

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

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

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

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

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

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

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

6.2.2 Identification

```

219  $\langle *base \rangle$ 
220 \NeedsTeXFormat{LaTeX2e}
221 \ProvidesPackage{zref-base}%
222 [2020-07-03 v2.32 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     }%
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     }%
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.

`\ZOE@propname`: Extract function.

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

`\ZC@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##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@wu@getcurrent#1{%
592 \etex@unexpanded\expandafter\expandafter\expandafter{%
593 \ZREF@getcurrent{#1}%
594 }%
595 }

```

`\zref@getcurrent`

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

6.2.8 Reference generation

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

```

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

```

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

```

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

```

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

```

610 \ZREF@Robust\def\zref@labelbyprops#1#2{%
611 \@bsphack
612 \begingroup
613 \toks@{}%
614 \comma@parse{#2}{%
615 \zref@ifpropundefined\comma@entry{%
616 \PackageWarning\ZREF@name{%
617 Property ‘\comma@entry’ is not known%
618 }%
619 }{%
620 \toks@\expandafter{%
621 \the\expandafter\toks@\csname\comma@entry\endcsname
622 }%
623 }%
624 \ltx@gobble
625 }%
626 \expandafter\endgroup
627 \expandafter\ZREF@label\expandafter{\the\toks@}{#1}%
628 \@esphack
629 }

```

`\zref@labelbykv`

```

630 \ZREF@Robust\def\zref@labelbykv#1#2{%
631 \@bsphack
632 \begingroup
633 \let\Z@L@ZREF@temp\ltx@empty
634 \kvsetkeys{ZREF@LABEL}{#1}%
635 \ifZREF@immediate

```

```

636     \expandafter\zref@wrapper@immediate\expandafter{%
637     \expandafter\ZREF@label\expandafter{\Z@L@ZREF@temp}#{2}%
638     }%
639     \else
640     \expandafter\ZREF@label\expandafter{\Z@L@ZREF@temp}#{2}%
641     \fi
642 \endgroup
643 \@esphack
644 }

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

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

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

672 \kv@define@key{ZREF@LABEL}{immediate}[true]{%
673 \edef\ZREF@param{#1}%
674 \ifx\ZREF@param\ZREF@true
675 \ZREF@immediatetrue
676 \else
677 \ifx\ZREF@param\ZREF@false
678 \ZREF@immediatefalse
679 \else
680 \PackageWarning\ZREF@name{%
681 Option 'immediate' expects 'true' or 'false'.\MessageBreak
682 Ignoring invalid value '\ZREF@param'%
683 }%
684 \fi
685 \fi
686 }

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

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

```

```

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

```

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

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

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

```

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

```

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

```

710 \def\ZREF@label#1#2{%
711   \if@filesw
712     \begingroup
713     \ifZREF@immediate
714       \let\ZREF@org@thepage\thepage
715     \fi
716     \protected@write\@auxout{%
717       \ifZREF@immediate
718         \let\thepage\ZREF@org@thepage
719       \fi
720       \let\ZREF@temp\ltx@empty
721       \@tfor\ZREF@P:=#1\do{%
722         \begingroup
723         \escapechar=-1 %
724         \edef\x{\endgroup
725           \def\noexpand\ZREF@P{%
726             \expandafter\string\ZREF@P
727           }%
728         }%
729         \x
730       \expandafter\ifx
731         \csname
732           \ifZREF@immediate

```

```

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

```

6.2.9 Reference querying and extracting

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

- In standard \LaTeX undefined references sets a flag and generate a warning. Both actions are not expandable.
- Babel's support for its shorthand uses commands that use non-expandable assignments. However currently there is hope, that primitives are added to pdf \TeX 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 `\zref@ifrefundefined` 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\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@activetrue\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.

```

909 \ZREF@Robust\def\zref@require@unique{%
910   \@ifundefined{c@zref@unique}{%
911     \begingroup
912     \let\@addtoreset\ltx@gobbletwo
913     \newcounter{zref@unique}%
914     \endgroup

```

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

```

915   \renewcommand*{\thezref@unique}{%
916     zref@number\c@zref@unique
917   }%
918 }{}%
919 }

```

6.2.12 Utilities

`\ZREF@number`

```

920 \ltx@ifundefined{numexpr}{%
921   \def\ZREF@number#1{\number#1}%
922 }{%
923   \def\ZREF@number#1{\the\numexpr(#1)\relax}%
924 }

```

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`.

```

925 \ZREF@Robust\def\zref@setdefault#1{%
926   \def\zref@default{#1}%
927 }

```

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

```

928 \zref@setdefault{%
929   \nfss@text{\reset@font\bfseries ??}%
930 }

```

Main property list.

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

```
931 \ZREF@Robust\def\zref@setmainlist#1{%
932   \def\ZREF@mainlist{#1}%
933 }
934 \zref@setmainlist{main}
```

Now we create the list.

```
935 \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

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

Properties

`\ZREF@NewPropAnchor`

```
939 \def\ZREF@NewPropAnchor{%
940   \zref@newprop{anchor}{%
941     \ltx@ifundefined{@currentHref}{\@currentHref}%
942   }%
943   \global\let\ZREF@NewPropAnchor\relax
944 }
```

`\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`

```
945 \def\ZREF@NewPropTitle{%
946   \gdef\zref@titleref@current{}%
947   \zref@newprop{title}{\zref@titleref@current}%
948   \global\let\ZREF@NewPropTitle\relax
949 }
```

`\ZREF@NewPropTheotype`

```
950 \def\ZREF@NewPropTheotype{%
951   \zref@newprop{theotype}{}%
952   \global\let\ZREF@NewPropTheotype\relax
953 }
```

`\ZREF@NewPropPageValue`

```
954 \def\ZREF@NewPropPageValue{%
955   \zref@newprop*{pagevalue}[0]{\number\c@page}%
956   \global\let\ZREF@NewPropPageValue\relax
957 }
```

Mark successful loading

```
958 \let\ZREF@base@ok=Y
959 /base)
```

6.3 Module user

```
960 (*user)
961 \NeedsTeXFormat{LaTeX2e}
962 \ProvidesPackage{zref-user}%
963 [2020-07-03 v2.32 Module user for zref (HO)]%
964 \RequirePackage{zref-base}[2019/11/29]
965 \ifx\ZREF@base@ok Y%
966 \else
967 \expandafter\endinput
968 \fi
```

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

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

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

```
969 \newcommand*\zlabel{%
970   \ifx\label\ltx@gobble
971     \expandafter\ltx@gobble
972   \else
973     \expandafter\zref@wrapper@babel\expandafter\zref@label
974   \fi
975 }%
```

`\zkvlabel`

```
976 \newcommand*\zkvlabel}[1]{%
977   \ifx\label\ltx@gobble
978     \expandafter\ltx@gobblethree
979   \fi
980   \zref@wrapper@babel{\zref@labelbykv{#1}}%
981 }%
```

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

```
982 \newcommand*\zref}[2][default]{% robust because of optional argument
983   \zref@propexists{#1}%
984   \zref@wrapper@babel\ZREF@zref{#2}{#1}%
985 }%
986 }%
987 \def\ZREF@zref#1{%
988   \zref@refused{#1}%
989   \zref@extract{#1}%
990 }%
```

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

```
991 \ZREF@IfDefinable\zpageref\def{%
992   {\zref[page]}%
993 }
```

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

```
994 \ZREF@IfDefinable\zrefused\def{%
995   {\zref@refused}%
996 }
997 \end{user}
```

6.4 Module `abspage`

```
998 (*abspage)
999 \NeedsTeXFormat{LaTeX2e}
1000 \ProvidesPackage{zref-abspage}%
1001 [2020-07-03 v2.32 Module abspage for zref (HO)]%
1002 \RequirePackage{zref-base}[2019/11/29]
1003 \ifx\ZREF@base@ok Y%
1004 \else
1005 \expandafter\endinput
1006 \fi
```

Module `abspage` adds a new property `abspage` to the main property list for absolute page numbers. These are recorded by the help of package `atbegshi`.

```
1007 \RequirePackage{atbegshi}[2011/10/05]%
```

The counter `abspage` must not go in the clear list of `@ckpt` that is used to set counters in `.aux` files of included `TeX` files.

```
1008 \begingroup
1009 \let\@addtoreset\ltx@gobbletwo
1010 \newcounter{abspage}%
1011 \endgroup
1012 \setcounter{abspage}{0}%
1013 \AtBeginShipout{%
1014 \stepcounter{abspage}%
1015 }%
1016 \zref@newprop*{abspage}[0]{\the\c@abspage}%
1017 \zref@addprop\ZREF@mainlist{abspage}%
```

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

```
1018 </abspage>
```

6.5 Module `counter`

```
1019 (*counter)
1020 \NeedsTeXFormat{LaTeX2e}
1021 \ProvidesPackage{zref-counter}%
1022 [2020-07-03 v2.32 Module counter for zref (HO)]%
1023 \RequirePackage{zref-base}[2019/11/29]
1024 \ifx\ZREF@base@ok Y%
1025 \else
1026 \expandafter\endinput
1027 \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.

```
1028 \@ifl@t@r\fmtversion{2020-10-01}
1029 {
1030 \zref@newprop{counter}{\@currentcounter}
1031 \zref@addprop\ZREF@mainlist{counter}
1032 }
1033 {
1034 \zref@newprop{counter}{}
1035 \zref@addprop\ZREF@mainlist{counter}
1036 \AtBeginDocument{%
1037 \ZREF@patch{refstepcounter}{%
```

```

1038     \def\refstepcounter#1{%
1039         \zref@setcurrent{counter}{#1}%
1040         \ZREF@org@refstepcounter{#1}%
1041     }%
1042 }%
1043 }
1044 }
1045 \end{counter}

```

6.6 Module lastpage

```

1046 (*lastpage)
1047 \NeedsTeXFormat{LaTeX2e}
1048 \ProvidesPackage{zref-lastpage}%
1049 [2020-07-03 v2.32 Module lastpage for zref (HO)]%
1050 \RequirePackage{zref-base}[2019/11/29]
1051 \RequirePackage{zref-abspage}[2019/11/29]
1052 \RequirePackage{atveryend}[2009/12/07]
1053 \ifx\ZREF@base@ok Y%
1054 \else
1055     \expandafter\endinput
1056 \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.

```

1057 \zref@newlist{LastPage}
1058 \AfterLastShipout{%
1059     \if@filesw
1060         \begingroup
1061             \advance\c@page\m@ne
1062             \toks@\expandafter\expandafter\expandafter{%
1063                 \expandafter\Z@L@main
1064                 \Z@L@LastPage
1065             }%
1066             \expandafter\zref@wrapper@immediate\expandafter{%
1067                 \expandafter\ZREF@label\expandafter{\the\toks@}{LastPage}%
1068             }%
1069         \endgroup
1070     \fi
1071 }

```

`\zref@iflastpage`

```

1072 \def\zref@iflastpage#1{%
1073     \ifnum\zref@extractdefault{#1}{abspage}{-1}=%
1074         \zref@extractdefault{LastPage}{abspage}{-2} %
1075     \expandafter\ltx@firstoftwo
1076     \else
1077     \expandafter\ltx@secondoftwo
1078     \fi
1079 }

```

`\ziflastpage`

```

1080 \ZREF@IfDefinable\ziflastpage\def{%
1081     {\zref@wrapper@babel\ZREF@iflastpage}%
1082 }

```

`ZREF@iflastpage`

```

1083 \def\ZREF@iflastpage#1{%
1084   \zref@refused{LastPage}%
1085   \zref@refused{#1}%
1086   \zref@iflastpage{#1}%
1087 }

1088 </lastpage>

```

6.7 Module thepage

```

1089 <*thepage>
1090 \NeedsTeXFormat{LaTeX2e}
1091 \ProvidesPackage{zref-thepage}%
1092   [2020-07-03 v2.32 Module thepage for zref (HO)]%
1093 \RequirePackage{zref-base}[2019/11/29]
1094 \ifx\ZREF@base@ok Y%
1095 \else
1096   \expandafter\endinput
1097 \fi

1098 \RequirePackage{atbegshi}[2011/10/05]
1099 \RequirePackage{zref-abspage}[2019/11/29]

1100 \zref@newlist{thepage}
1101 \zref@addprop{thepage}{page}
1102 \ZREF@NewPropPageValue

```

\zref@thepage@atbegshi@hook

```

1103 \let\zref@thepage@atbegshi@hook\ltx@empty

1104 \zref@addprop{thepage}{pagevalue}
1105 \AtBeginShipout{%
1106   \AtBeginShipoutAddToBox{%
1107     \zref@thepage@atbegshi@hook
1108     \zref@labelbylist{thepage\the\value{abspage}}{thepage}%
1109   }%
1110 }

```

\zref@thepage@name

```

1111 \ltx@ifundefined{numexpr}{%
1112   \def\zref@thepage@name#1{thepage\number#1}%
1113 }{%
1114   \def\zref@thepage@name#1{thepage\the\numexpr#1}%
1115 }

```

\zref@thepage

```

1116 \def\zref@thepage#1{%
1117   \zref@extract{\zref@thepage@name{#1}}{page}%
1118 }%

```

\zref@thepage@refused

```

1119 \ZREF@Robust\def\zref@thepage@refused#1{%
1120   \zref@refused{\zref@thepage@name{#1}}%
1121 }%

```

\zthepage

```

1122 \ZREF@ifdefinable\zthepage\def{%
1123   #1{%
1124     \zref@thepage@refused{#1}%

```

```

1125   \zref@thepage{#1}%
1126 }%
1127 }

1128 </thepage>

```

6.8 Module nextpage

```

1129 (*nextpage)
1130 \NeedsTeXFormat{LaTeX2e}
1131 \ProvidesPackage{zref-nextpage}%
1132 [2020-07-03 v2.32 Module nextpage for zref (HO)]%
1133 \RequirePackage{zref-base}[2019/11/29]
1134 \ifx\ZREF@base@ok Y%
1135 \else
1136   \expandafter\endinput
1137 \fi

1138 \RequirePackage{zref-abspage}[2019/11/29]
1139 \RequirePackage{zref-thepage}[2019/11/29]
1140 \RequirePackage{zref-lastpage}[2019/11/29]
1141 \RequirePackage{uniquecounter}[2009/12/18]

1142 \UniqueCounterNew{znextpage}
1143
1144 \newcommand*{\znextpagesetup}{%
1145   \afterassignment\ZREF@np@setup@i
1146   \def\ZREF@np@call@unknown##1%
1147 }
1148 \def\ZREF@np@setup@i{%
1149   \afterassignment\ZREF@np@setup@ii
1150   \def\ZREF@np@call@nonext##1%
1151 }
1152 \def\ZREF@np@setup@ii{%
1153   \def\ZREF@np@call@next##1%
1154 }
1155 \def\ZREF@np@call@unknown#1{#1}
1156 \def\ZREF@np@call@nonext#1{#1}
1157 \def\ZREF@np@call@next#1{#1}
1158 \ZREF@IfDefinable\znextpage\def{%
1159   {\UniqueCounterCall{znextpage}{\ZREF@nextpage}}%
1160 }%
1161 \newcommand*{\znonextpagename}{}
1162 \newcommand*{\zunknownnextpagename}{\Z@D@page}
1163 \def\ZREF@nextpage#1{%
1164   \begingroup
1165     \def\ZREF@refname@this{zref@np#1}%
1166     \zref@labelbyprops\ZREF@refname@this{abspage}%
1167     \chardef\ZREF@call=0 % unknown
1168     \ZREF@ifrefundefined\ZREF@refname@this{%
1169       }{%
1170         \edef\ZREF@pagenum@this{%
1171           \zref@extractdefault\ZREF@refname@this{abspage}{0}%
1172         }%
1173         \edef\ZREF@refname@next{%
1174           \zref@thepage@name{%
1175             \the\numexpr\ZREF@pagenum@this+1%
1176           }%
1177         }%
1178         \ifnum\ZREF@pagenum@this>0 %

```



```

1179     \ZREF@ifrefundefined{LastPage}{%
1180     \zref@ifrefundefined\ZREF@refname@next{%
1181     }{%
1182     \chardef\ZREF@call=2 % next page
1183     }%
1184   }{%
1185     \edef\ZREF@pagenum@last{%
1186     \zref@extractdefault{LastPage}{abspage}{0}%
1187     }%
1188     \ifnum\ZREF@pagenum@this<\ZREF@pagenum@last\ltx@space
1189     \ZREF@ifrefundefined\ZREF@refname@next{%
1190     }{%
1191     \chardef\ZREF@call=2 % next page
1192     }%
1193   \else
1194     \ifnum\ZREF@pagenum@this=\ZREF@pagenum@this\ltx@space
1195     \chardef\ZREF@call=1 % no next page
1196     \fi
1197   \fi
1198   }%
1199 \fi
1200 }%
1201 \edef\x{%
1202 \endgroup
1203 \ifcase\ZREF@call
1204 \noexpand\ZREF@np@call@unknown{%
1205 \noexpand\zunknownnextpagename
1206 }%
1207 \or
1208 \noexpand\ZREF@np@call@nonext{%
1209 \noexpand\znonextpagename
1210 }%
1211 \else
1212 \noexpand\ZREF@np@call@next{%
1213 \noexpand\zref@extract{\ZREF@refname@next}{page}%
1214 }%
1215 \fi
1216 }%
1217 \x
1218 }
1219 </nextpage>

```

6.9 Module totpages

```

1220 (*totpages)
1221 \NeedsTeXFormat{LaTeX2e}
1222 \ProvidesPackage{zref-totpages}%
1223 [2020-07-03 v2.32 Module totpages for zref (HO)]%
1224 \RequirePackage{zref-base}[2019/11/29]
1225 \ifx\ZREF@base@ok Y%
1226 \else
1227 \expandafter\endinput
1228 \fi

```

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

```

1229 \RequirePackage{zref-abspage}[2019/11/29]
1230 \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.

```

1231 \newcommand*\ztotpages}{%
1232   \zref@extractdefault{LastPage}{abspage}{0}%
1233 }

```

Also we mark the reference LastPage as used:

```

1234 \AtBeginDocument{%
1235   \zref@refused{LastPage}%
1236 }
1237 </totpages>

```

6.10 Module pagelayout

```

1238 (*pagelayout)
1239 \NeedsTeXFormat{LaTeX2e}
1240 \ProvidesPackage{zref-pagelayout}%
1241   [2020-07-03 v2.32 Module pagelayout for zref (HO)]%
1242 \RequirePackage{zref-base}[2019/11/29]
1243 \ifx\ZREF@base@ok Y%
1244 \else
1245   \expandafter\endinput
1246 \fi
1247 \RequirePackage{zref-thepage}[2019/11/29]
1248 \RequirePackage{iftex}[2019/11/07]%
1249 \RequirePackage{atveryend}[2010/03/24]

```

6.10.1 Define layout properties

```

1250 \def\ZREF@temp#1{%
1251   \begingroup
1252     \escapechar=-1 %
1253     \ltx@ifundefined{string#1}{\endgroup}{%
1254       \edef\x{%
1255         \endgroup
1256         \noexpand\zref@newprop*{\string#1}%
1257           [\noexpand\number\noexpand#1]% hash-ok
1258           {\noexpand\number\noexpand#1}%
1259         \noexpand\zref@addprop{thepage}{\string#1}%
1260       }%
1261       \x
1262     }%
1263 }
1264 \ZREF@temp\mag
1265 \ZREF@temp\paperwidth
1266 \ZREF@temp\paperheight
1267 \ZREF@temp\stockwidth % memoir.cls, crop.sty
1268 \ZREF@temp\stockheight % memoir.cls, crop.sty
1269 \ZREF@temp\mediawidth % VTeX
1270 \ZREF@temp\mediaheight % VTeX
1271 \ifluatex
1272 \zref@newprop*{pdfvorigin}%
1273   [\number\pdfvariable vorigin]% hash-ok
1274   {\number\pdfvariable vorigin}%
1275 \zref@addprop{thepage}{pdfvorigin}
1276 \zref@newprop*{pdfhorigin}%
1277   [\number\pdfvariable horigin]% hash-ok
1278   {\number\pdfvariable horigin}%
1279 \zref@addprop{thepage}{pdfhorigin}
1280 \zref@newprop*{pdfpageheight}%

```

```

1281             [\number\pageheight]% hash-ok
1282             {\number\pageheight}%
1283 \zref@addprop{thepage}{pdfpageheight}
1284 \zref@newprop*{pdfpagewidth}%
1285             [\number\pagewidth]% hash-ok
1286             {\number\pagewidth}%
1287 \zref@addprop{thepage}{pdfpagewidth}
1288 \else
1289 \ZREF@temp\pdfpagewidth
1290 \ZREF@temp\pdfpageheight
1291 \ZREF@temp\pdfhorigin
1292 \ZREF@temp\pdfvorigin
1293 \fi
1294 \ZREF@temp\hoffset
1295 \ZREF@temp\voffset
1296 \ZREF@temp\topmargin
1297 \ZREF@temp\oddsidemargin
1298 \ZREF@temp\evensidemargin
1299 \ZREF@temp\textwidth
1300 \ZREF@temp\textheight
1301 \ZREF@temp\headheight
1302 \ZREF@temp\headsep
1303 \ZREF@temp\footskip
1304 \ZREF@temp\marginparwidth
1305 \ZREF@temp\marginparsep
1306 \ZREF@temp\columnwidth
1307 \ZREF@temp\columnsep
1308 \ZREF@temp\trimedge % memoir.cls
1309 \ZREF@temp\spinemargin % memoir.cls
1310 \ZREF@temp\foremargin % memoir.cls
1311 \ZREF@temp\trintop % memoir.cls
1312 \ZREF@temp\uppermargin % memoir.cls
1313 \ZREF@temp\headmargin % memoir.cls
1314 \IfFormatAtLeastTF{2020/10/01}
1315 {
1316   \zref@newprop*{outputboxwd}[Opt]{\ShipoutBoxWidth}
1317   \zref@newprop*{outputboxht}[Opt]{\ShipoutBoxHeight}
1318   \zref@newprop*{outputboxdp}[Opt]{\ShipoutBoxDepth}
1319 }
1320 {
1321   \zref@newprop*{outputboxwd}[Opt]{\AtBeginShipoutBoxWidth}
1322   \zref@newprop*{outputboxht}[Opt]{\AtBeginShipoutBoxHeight}
1323   \zref@newprop*{outputboxdp}[Opt]{\AtBeginShipoutBoxDepth}
1324 }
1325 \zref@addprops{thepage}{outputboxwd,outputboxht,outputboxdp}
\ifZREF@pl@list
1326 \ltx@newif\ifZREF@pl@list
\zref@listpagelayout
1327 \ZREF@IfDefinable\zlistpagelayout\def{%
1328   {\global\ZREF@pl@listtrue}%
1329 }
\ZREF@pl@AfterLastShipout
1330 \def\ZREF@pl@AfterLastShipout{%
1331   \ifZREF@pl@list

```

```

1332 \edef\ZREF@page@max{\the\value{abspage}}%
1333 \ltx@ifundefined{ZREF@org@testdef}{%
1334 \let\ZREF@org@testdef\@testdef
1335 \def\@testdef##1##2##3{%
1336 \ZREF@org@testdef{##1}{##2}{##3}%
1337 \def\ZREF@temp{##1}%
1338 \ifx\ZREF@temp\ZREF@RefPrefix
1339 \expandafter\gdef\csname##1@##2\endcsname{##3}%
1340 \fi
1341 }%
1342 }{%%
1343 \AtVeryEndDocument{\ZREF@pl@AtVeryEnd}%
1344 \fi
1345 }

```

\ZREF@pl@AtVeryEnd

```

1346 \def\ZREF@pl@AtVeryEnd{%
1347 \begingroup
1348 \toks@{Page layout parameters:\MessageBreak}%
1349 \count@=1 %
1350 \ZREF@pl@ListPage
1351 \edef\x{\endgroup
1352 \noexpand\@PackageInfoNoLine{zref-pagelayout}{\the\toks@}%
1353 }%
1354 \x
1355 }

```

\ZREF@pl@ListPage

```

1356 \def\ZREF@pl@ListPage{%
1357 \edef\x{%
1358 \toks@={%
1359 \the\toks@
1360 Page \the\count@:\noexpand\MessageBreak
1361 \zref@ifrefundefined{thepage\the\count@}{}{%
1362 \ltx@space\ltx@space mag = %
1363 \zref@extract{thepage\the\count@}{mag}%
1364 \noexpand\MessageBreak
1365 \ZREF@pl@ListEntry{paperwidth}%
1366 \ZREF@pl@ListEntry{paperheight}%
1367 \ZREF@pl@ListEntry{stockwidth}%
1368 \ZREF@pl@ListEntry{stockheight}%
1369 \ZREF@pl@ListEntry{mediawidth}%
1370 \ZREF@pl@ListEntry{mediaheight}%
1371 \ZREF@pl@ListEntry{pdfpagewidth}%
1372 \ZREF@pl@ListEntry{pdfpageheight}%
1373 \ZREF@pl@ListEntry{pdfhorigin}%
1374 \ZREF@pl@ListEntry{pdfvorigin}%
1375 \ZREF@pl@ListEntry{hoffset}%
1376 \ZREF@pl@ListEntry{voffset}%
1377 \ZREF@pl@ListEntry{topmargin}%
1378 \ZREF@pl@ListEntry{oddsidemargin}%
1379 \ZREF@pl@ListEntry{evensidemargin}%
1380 \ZREF@pl@ListEntry{textwidth}%
1381 \ZREF@pl@ListEntry{textheight}%
1382 \ZREF@pl@ListEntry{headheight}%
1383 \ZREF@pl@ListEntry{headsep}%
1384 \ZREF@pl@ListEntry{footskip}%
1385 \ZREF@pl@ListEntry{marginparwidth}%

```

```

1386     \ZREF@pl@ListEntry{marginparsep}%
1387     \ZREF@pl@ListEntry{columnwidth}%
1388     \ZREF@pl@ListEntry{columnsep}%
1389     \ZREF@pl@ListEntry{trimedge}%
1390     \ZREF@pl@ListEntry{spinemargin}%
1391     \ZREF@pl@ListEntry{foremargin}%
1392     \ZREF@pl@ListEntry{trimtop}%
1393     \ZREF@pl@ListEntry{uppermargin}%
1394     \ZREF@pl@ListEntry{headmargin}%
1395   }%
1396 }%
1397 }\x
1398 \ifnum\ZREF@page@max>\count@
1399   \advance\count@ by\ltx@one
1400 \else
1401   \expandafter\ltx@gobble
1402 \fi
1403 \ZREF@pl@ListPage
1404 }

```

\ZREF@pl@ListEntry

```

1405 \def\ZREF@pl@ListEntry#1{%
1406   \zref@ifpropundefined{#1}{%
1407   }{%
1408     \zref@ifrefcontainsprop{thepage\the\count@}{#1}{%
1409       \ltx@space\ltx@space#1 = %
1410       \zref@extract{thepage\the\count@}{#1}sp = %
1411       \the\dimexpr\zref@extract{thepage\the\count@}{#1}sp\relax
1412       \noexpand\MessageBreak
1413     }{%
1414     }%
1415   }

```

```

1416 \AfterLastShipout{%
1417   \ZREF@pl@AfterLastShipout
1418 }

```

```

1419 </pagelayout>

```

6.11 Module pageattr

```

1420 (*pageattr)
1421 \NeedsTeXFormat{LaTeX2e}
1422 \ProvidesPackage{zref-pageattr}%
1423   [2020-07-03 v2.32 Module pageattr for zref (HO)]%
1424 \RequirePackage{zref-base}[2019/11/29]
1425 \ifx\ZREF@base@ok Y%
1426 \else
1427   \expandafter\endinput
1428 \fi

```

```

1429 \RequirePackage{iftex}[2019/11/07]%

```

```

1430 \let\ZREF@temp=N%
1431 \ifluatex
1432   \expandafter\@firstoftwo
1433 \else
1434   \expandafter\@secondoftwo
1435 \fi
1436 {\%luatex

```

```

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

1471 \ifx\ZREF@temp N%
1472 \expandafter\endinput
1473 \fi

1474 \RequirePackage{zref-abspage}[2019/11/29]
1475 \RequirePackage{atveryend}[2010/03/24]
1476 \RequirePackage{pdftexcmds}[2010/04/01]
1477 \let\ZREF@temp=Y%
1478 \ltx@ifundefined{pdf@escapehex}{\let\ZREF@temp=N}{%
1479 \ltx@ifundefined{pdf@unescapehex}{\let\ZREF@temp=N}{%
1480 \ifx\ZREF@temp N%
1481   \let\zref@hex\ltx@firstofone
1482   \let\zref@unhex\ltx@firstofone
1483 \else
1484   \let\zref@hex\pdf@escapehex
1485   \let\zref@unhex\pdf@unescapehex
1486 \fi

\ifZREF@pa@list

1487 \ltx@newif\ifZREF@pa@list

\zref@listpageattr

1488 \ZREF@ifdefinable\zlistpageattr\def{%
1489   {\ZREF@pa@listtrue}%
1490 }

```

\ZREF@pa@AfterLastShipout

```
1491 \def\ZREF@pa@AfterLastShipout{%
1492   \ifZREF@pa@list
1493     \edef\ZREF@page@max{\the\value{abspage}}%
1494     \ltx@ifundefined{ZREF@org@testdef}{%
1495       \let\ZREF@org@testdef\@testdef
1496       \def\@testdef##1##2##3{%
1497         \ZREF@org@testdef{##1}-{##2}-{##3}%
1498         \def\ZREF@temp{##1}%
1499         \ifx\ZREF@temp\ZREF@RefPrefix
1500           \expandafter\xdef\csname##1@##2\endcsname{##3}%
1501         \fi
1502       }%
1503     }-%
1504     \AtVeryEndDocument{\ZREF@pa@AtVeryEnd}%
1505   \fi
1506 }
```

\ZREF@pa@AtVeryEnd

```
1507 \let\ZREF@temp=Y%
1508 \ltx@ifundefined{pdfpageattr}{\let\ZREF@temp=N}
1509 \ifluatex \let\ZREF@temp=N \fi
1510 \ifx\ZREF@temp Y
1511   \expandafter\@firstoftwo
1512 \else
1513   \expandafter\@secondoftwo
1514 \fi
1515 {%
1516   \def\ZREF@pa@AtVeryEnd{%
1517   }
1518   {%
1519     \def\ZREF@pa@AtVeryEnd{%
1520       \begingroup
1521         \toks@{List of \ltx@backslashchar
1522           \ifluatex pdfvariable\else pdf\fi
1523           pdfpageattr:\MessageBreak}%
1524         \count@=1 %
1525         \ZREF@pa@ListPage
1526         \edef\x{\endgroup
1527           \noexpand\@PackageInfoNoLine{zref-pageattr}{%
1528             \the\toks@
1529           }%
1530         }%
1531       \x
1532     }%
1533   }%
1534 }
```

\zref@pageattr

```
1533 \def\zref@pdfpageattr#1{%
1534   \zref@unhex{%
1535     \zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}%
1536   }%
1537 }
1538 % compability, \zref@pageattr was defined in older versions
1539 \let\zref@pageattr\zref@pdfpageattr
```

\zref@pageattr@used

```
1540 \ZREF@Robust\def\zref@pageattr@used#1{%
1541   \zref@refused{thepage\ZREF@number{#1}}%
```

1542 }

\ZREF@pa@ListPage

```
1543 \def\ZREF@pa@ListPage{%
1544   \edef\x{%
1545     \toks@={%
1546       \the\toks@
1547       Page \the\count@:%
1548       \noexpand\MessageBreak
1549       \zref@ifrefundefined{thepage\the\count@}{-}{%
1550         <<\zref@pdfpagesattr\count@>>%
1551       \noexpand\MessageBreak
1552     }%
1553   }%
1554 } \x
1555 \ifnum\ZREF@page@max>\count@
1556   \advance\count@ by\ltx@one
1557 \else
1558   \expandafter\ltx@gobble
1559 \fi
1560 \ZREF@pa@ListPage
1561 }%
```

1562 }

```
1563 \let\ZREF@temp=Y%
1564 \ltx@ifundefined{pdfpagesattr}{-}{\let\ZREF@temp=N}
1565 \ifluatex \let\ZREF@temp=N \fi
1566 \ifx\ZREF@temp N
1567   \expandafter\@firstofone
1568 \fi
1569 {%
```

\zref@pdfpagesattr

```
1570 \def\zref@pdfpagesattr{%
1571   \zref@unhex{%
1572     \zref@extract{LastPage}{pdfpagesattr}%
1573   }%
1574 }%
```

\zref@pdfpagesattr@used

```
1575 \ZREF@Robust\def\zref@pdfpagesattr@used{%
1576   \zref@refused{LastPage}%
1577 }%

1578 \ltx@LocalAppendToMacro\ZREF@pa@AtVeryEnd{%
1579   \@PackageInfoNoLine{zref-pageattr}{%
1580     \ltx@backslashchar
1581     \ifluatex pdfvariable\else pdf\fi
1582     pagesattr:\MessageBreak
1583     <<\zref@pdfpagesattr>>%
1584     \MessageBreak
1585   }%
1586 }%
1587 }
1588 \AfterLastShipout{%
1589   \ZREF@pa@AfterLastShipout
1590 }
1591 (/pageattr)
```


6.12 Module marks

```

1592 (*marks)
1593 \NeedsTeXFormat{LaTeX2e}
1594 \ProvidesPackage{zref-marks}%
1595 [2020-07-03 v2.32 Module marks for zref (HO)]%
1596 \RequirePackage{zref-base}[2019/11/29]
1597 \ifx\ZREF@base@ok Y%
1598 \else
1599 \expandafter\endinput
1600 \fi

1601 \newcommand*{\zref@marks@register}[3] [] {%
1602 \edef\ZREF@TempName{#1}%
1603 \edef\ZREF@TempNum{\ZREF@number{#2}}%
1604 \ifnum\ZREF@TempNum<\ltx@zero %
1605 \PackageError\ZREF@name{%
1606 \string\zref@marks@register\ltx@space is called with invalid%
1607 \MessageBreak
1608 marks register number (\ZREF@TempNum)%
1609 }{%
1610 Use ‘0’ or the command, defined by \string\newmarks.\MessageBreak
1611 \@ehc
1612 }%
1613 \else
1614 \ifx\ZREF@TempName\ltx@empty
1615 \edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}%
1616 \else
1617 \edef\ZREF@TempName{marks\ZREF@TempName}%
1618 \fi
1619 \ZREF@MARKS@DefineProp{top}%
1620 \ZREF@MARKS@DefineProp{first}%
1621 \ZREF@MARKS@DefineProp{bot}%
1622 \kv@parse{#3}{%
1623 \ifx\kv@value\relax
1624 \def\kv@value{top,first,bot}%
1625 \fi
1626 \edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}%
1627 \ifx\ZREF@temp\ZREF@STAR
1628 \edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}%
1629 \zref@newlist\kv@key
1630 \fi
1631 \expandafter\comma@parse\expandafter{\kv@value}{%
1632 \ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else
1633 \ifx\comma@entry\ZREF@NAME@first 1\else
1634 \ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space
1635 \PackageWarning{zref-marks}{%
1636 Use ‘top’, ‘first’ or ‘bot’ for the list values%
1637 \MessageBreak
1638 in the third argument of \string\zref@marks@register.%
1639 \MessageBreak
1640 Ignoring unkown value ‘\comma@entry’%
1641 }%
1642 \else
1643 \zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}%
1644 \fi
1645 \ltx@gobble
1646 }%
1647 \ltx@gobbletwo

```

```

1648   }%
1649   \fi
1650 }
1651 \def\ZREF@STAR{*}
1652 \def\ZREF@NAME@top{top}
1653 \def\ZREF@NAME@first{first}
1654 \def\ZREF@NAME@bot{bot}
1655 \def\ZREF@MARKS@DefineProp#1{%
1656   \zref@ifpropundefined{#1\ZREF@TempName}{%
1657     \ifnum\ZREF@TempNum=\ltx@zero
1658       \begingroup
1659         \edef\x{\endgroup
1660           \noexpand\zref@newprop*{#1\ZREF@TempName}[] {%
1661             \expandafter\noexpand\csname#1mark\endcsname
1662           }%
1663         }%
1664         \x
1665       \else
1666         \begingroup
1667         \edef\x{\endgroup
1668           \noexpand\zref@newprop*{#1\ZREF@TempName}[] {%
1669             \expandafter\noexpand\csname#1marks\endcsname
1670             \ZREF@TempNum
1671           }%
1672         }%
1673         \x
1674       \fi
1675   }{%
1676     \PackageWarning{zref-marks}{%
1677       \string\zref@marks@register\ltx@space does not generate the%
1678       \MessageBreak
1679       new property ‘#1\ZREF@TempName’, because\MessageBreak
1680       it is already defined%
1681     }%
1682   }%
1683 }
1684 </marks>

```

6.13 Module runs

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

```

1685 <*runs>
1686 \NeedsTeXFormat{LaTeX2e}
1687 \ProvidesPackage{zref-runs}%
1688 [2020-07-03 v2.32 Module runs for zref (H0)]%

```

\backslash zruns

```

1689 \providecommand*\zruns{0}%
1690 \AtBeginDocument{%
1691   \edef\zruns{\number\numexpr\zruns+1}%
1692   \begingroup
1693     \def\on@line{}%
1694     \PackageInfo{zref-runs}{LaTeX runs: \zruns}%
1695     \if@filesw
1696       \immediate\write\@mainaux{%
1697         \string\gdef\string\zruns{\zruns}%
1698       }%

```

```

1699   \fi
1700 \endgroup
1701 }

1702 </runs>

```

6.14 Module perpage

```

1703 (*perpage)
1704 \NeedsTeXFormat{LaTeX2e}
1705 \ProvidesPackage{zref-perpage}%
1706 [2020-07-03 v2.32 Module perpage for zref (HO)]%
1707 \RequirePackage{zref-base}[2019/11/29]
1708 \ifx\ZREF@base@ok Y%
1709 \else
1710   \expandafter\endinput
1711 \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.

```

1712 \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.

```

1713 \ZREF@NewPropPageValue
1714 \zref@newlist{perpage}
1715 \zref@addprops{perpage}{abspage,page,pagevalue}

```

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

```

1716 \newcounter{zpage}

```

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

```

1717 \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`.

```

1718 \let\ZREF@org@stepcounter\stepcounter
1719 \def\stepcounter#1{%
1720   \ifcsname @stepcounterhook@#1\endcsname
1721     \csname @stepcounterhook@#1\endcsname
1722   \fi
1723   \ZREF@org@stepcounter{#1}%
1724 }

```

`\@stpelt` must be adapted due to the change in latex 2015-01, see <https://github.com/ho-tex/zref/issues/26>

```

1725 \let\ZREF@org@@stpelt\@stpelt
1726 \def\@stpelt#1{%
1727   \ifcsname ZREF@perpage@#1\endcsname
1728     \begingroup
1729       \let\stepcounter\ZREF@org@stepcounter
1730       \ZREF@org@@stpelt{#1}%
1731     \endgroup
1732     \expandafter\ltx@gobbletwo
1733   \fi
1734   \ZREF@org@@stpelt{#1}%
1735 }

```

`\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.

```

1736 \ZREF@ifDefinable\zmakeperpage\def{%
1737   {%
1738     \@ifnextchar[\ZREF@makeperpage@opt{\ZREF@@makeperpage[\ltx@zero]}%
1739   }%
1740 }

```

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

```

1741 \def\ZREF@makeperpage@opt[#1]{%
1742   \begingroup
1743   \edef\x{\endgroup
1744     \noexpand\ZREF@@makeperpage[\number\numexpr#1-1\relax]%
1745   }%
1746   \x
1747 }

1748 \def\ZREF@@makeperpage[#1]#2{%
1749   \@ifundefined{stepcounterhook@#2}{%
1750     \expandafter\gdef\csname @stepcounterhook@#2\endcsname{%
1751   }%
1752   \expandafter\gdef\csname ZREF@perpage@#2\endcsname{%
1753     \ZREF@@perpage@step{#2}{#1}%
1754   }%
1755   \expandafter\g@addto@macro\csname @stepcounterhook@#2\endcsname{%
1756     \ifcsname ZREF@perpage@#2\endcsname
1757       \csname ZREF@perpage@#2\endcsname
1758     \fi
1759   }%
1760 }

```

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

```

1761 \def\ZREF@@perpage@step#1#2{%

```

First the reference is generated.

```

1762   \global\advance\c@zref@unique\ltx@one
1763   \begingroup
1764   \expandafter
1765   \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.

```

1766   \zref@ifrefundefined\thezref@unique{%
1767     \global\c@zpage=\c@page
1768     \global\let\thezpage\thepage
1769     \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%
1770       \number\c@abspage
1771     }%
1772   }%

```

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

```

1773     \global\c@zpage=\zref@extract\thezref@unique{pagevalue}\relax
1774     \xdef\thezpage{\noexpand\zref@extract{\thezref@unique}{page}}%
1775     \expandafter\xdef\csname ZREF@abspage@#1\endcsname{%

```

```

1776     \zref@extractdefault\thezref@unique
1777         {abspage}{\number\c@abspage}%
1778     }%
1779 }%

```

Page changes are detected by a changed absolute page number.

```

1780 \expandafter\ifx\csname ZREF@abspage@#1\expandafter\endcsname
1781     \csname ZREF@currentabspage@#1\endcsname
1782 \else
1783     \global\csname c@#1\endcsname=#2\relax
1784     \global\expandafter\let
1785         \csname ZREF@currentabspage@#1\expandafter\endcsname
1786         \csname ZREF@abspage@#1\endcsname
1787 \fi
1788 \endgroup
1789 }

```

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

```

1790 \ZREF@ifDefinable\zunmakeperpage\def{%
1791     #1{%
1792     \global\expandafter
1793     \let\csname ZREF@perpage@#1\endcsname\@undefined
1794     }%
1795 }

```

1796 `/perpage`

6.15 Module `titleref`

```

1797 (*titleref)
1798 \NeedsTeXFormat{LaTeX2e}
1799 \ProvidesPackage{zref-titleref}%
1800 [2020-07-03 v2.32 Module titleref for zref (HO)]%
1801 \RequirePackage{zref-base}[2019/11/29]
1802 \ifx\ZREF@base@ok Y%
1803 \else
1804     \expandafter\endinput
1805 \fi
1806 \RequirePackage{getttitlestring}[2009/12/08]

```

6.15.1 Implementation

```
1807 \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.

```

1808 \ZREF@NewPropTitle
1809 \zref@addprop\ZREF@mainlist{title}%

```

The title strings go into the `.aux` file, thus they need some kind of protection. Package `titleref` uses a protected expansion method. The advantage is that this can be used to cleanup the string and to remove `\label`, `\index` and other macros unwanted for referencing. But there is the risk that fragile stuff can break.

Therefore package `nameref` does not expand the string. Thus the entries can safely be written to the `.aux` file. But potentially dangerous macros such as `\label` remain in the string and can cause problems when using the string in references.

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

```

1810 \newif\ifzref@titleref@expand

\ZREF@titleref@hook The hook \ZREF@titleref@hook allows to extend the cleanup for the expansion
method. Thus unnecessary macros can be removed or dangerous commands re-
moved. The hook is executed before the expansion of \zref@titleref@current.
1811 \let\ZREF@titleref@hook\ltx@empty

\zref@titleref@cleanup The hook should not be used directly, instead we provide the macro
\zref@titleref@cleanup to add stuff to the hook and prevents that a previous
non-empty content is not discarded accidently.
1812 \ZREF@Robust\def\zref@titleref@cleanup#1{%
1813   \begingroup
1814   \toks@\expandafter{%
1815     \ZREF@titleref@hook
1816     #1%
1817   }%
1818   \expandafter\endgroup
1819   \expandafter\def\expandafter\ZREF@titleref@hook\expandafter{%
1820     \the\toks@
1821   }%
1822 }%

\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.
1823 \newif\ifzref@titleref@stripperperiod
1824 \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.
1825 \ZREF@Robust\def\zref@titleref@setcurrent#1{%
1826   \ifzref@titleref@expand
1827     \GetTitleStringExpand{#1}%
1828   \else
1829     \GetTitleStringNonExpand{#1}%
1830   \fi
1831   \edef\zref@titleref@current{%
1832     \detokenize\expandafter{\GetTitleStringResult}%
1833   }%
1834   \ifzref@titleref@stripperperiod
1835     \edef\zref@titleref@current{%
1836       \expandafter\ZREF@stripperperiod\zref@titleref@current
1837       \ltx@empty.\ltx@empty\@nil
1838     }%
1839   \fi
1840 }%
1841 \GetTitleStringDisableCommands{%
1842   \ZREF@titleref@hook
1843 }

\ZREF@stripperperiod If \ZREF@stripperperiod is called, the argument consists of space tokens and tokens
with catcode 12 (other), because of  $\epsilon$ -TeX's \detokenize.
1844 \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`.

```
1845 \define@key{ZREF@TR}{expand}[true]{%
1846   \csname zref@titleref@expand#1\endcsname
1847 }%
1848 \define@key{ZREF@TR}{stripperperiod}[true]{%
1849   \csname zref@titleref@stripperperiod#1\endcsname
1850 }%
1851 \define@key{ZREF@TR}{cleanup}{%
1852   \zref@titleref@cleanup#1}%
1853 }%
1854 \define@key{ZREF@TR}{title}{%
1855   \def\zref@titleref@current#1}%
1856 }%
1857 \ZREF@IfDefinable\ztitlerefsetup\def{%
1858   {\kvsetkeys{ZREF@TR}}%
1859 }%
```

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

```
1860 \ZREF@IfDefinable\ztitleref\def{%
1861   {\zref@wrapper@babel\ZREF@titleref}%
1862 }%
1863 \def\ZREF@titleref#1{%
1864   \begingroup
1865     \zref@refused#1}%
1866     \let\label\tx@gobble
1867     \zref@extract#1{title}%
1868   \endgroup
1869 }%
```

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.

```
1870 \AtBeginDocument{%
1871   \ZREF@patch{@caption}{%
1872     \long\def@caption#1[#2]{%
1873       \zref@titleref@setcurrent#2}%
1874       \ZREF@org@@caption#1[#2]}%
1875   }%
1876 }%
```

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

```
1877 \ZREF@patch{@part}{%
1878   \def@part#1{%
1879     \zref@titleref@setcurrent#1}%
1880     \ZREF@org@@part[#1]}%
1881   }%
1882 }%
1883 \ZREF@patch{@chapter}{%
1884   \def@chapter#1{%
1885     \zref@titleref@setcurrent#1}%
1886     \ZREF@org@@chapter[#1]}%
1887   }%
1888 }%
```

```

1887   }%
1888 }%
1889 \ZREF@patch{@sect}{%
1890   \def\@sect#1#2#3#4#5#6[#7]{%
1891     \zref@titleref@setcurrent{#7}%
1892     \ZREF@org@@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]%
1893   }%
1894 }%

```

The star versions of the section commands.

```

1895 \ZREF@patch{@spart}{%
1896   \def\@spart#1{%
1897     \zref@titleref@setcurrent{#1}%
1898     \ZREF@org@@spart{#1}%
1899   }%
1900 }%
1901 \ZREF@patch{@schapter}{%
1902   \def\@schapter#1{%
1903     \zref@titleref@setcurrent{#1}%
1904     \ZREF@org@@schapter{#1}%
1905   }%
1906 }%
1907 \ZREF@patch{@ssect}{%
1908   \def\@ssect#1#2#3#4#5{%
1909     \zref@titleref@setcurrent{#5}%
1910     \ZREF@org@@ssect{#1}{#2}{#3}{#4}{#5}%
1911   }%
1912 }%

```

6.15.4 Environment description

```

1913 \ZREF@patch{descriptionlabel}{%
1914   \def\descriptionlabel#1{%
1915     \zref@titleref@setcurrent{#1}%
1916     \ZREF@org@descriptionlabel{#1}%
1917   }%
1918 }%

```

6.15.5 Class memoir

```

1919 \@ifclassloaded{memoir}{%
1920   \ltx@ifundefined{ifheadnameref}{}{%
1921     \def\@chapter[#1]#2{%
1922       \ltx@ifundefined{ch@pt@c}{%
1923         \zref@titleref@setcurrent{#1}%
1924       }{%
1925         \ifx\ch@pt@c\ltx@empty
1926           \zref@titleref@setcurrent{#2}%
1927         \else
1928           \def\NR@temp{#1}%
1929           \ifx\NR@temp\ltx@empty
1930             \expandafter\zref@titleref@setcurrent
1931             \expandafter{\ch@pt@c}%
1932           \else
1933             \ifheadnameref
1934               \zref@titleref@setcurrent{#1}%
1935             \else
1936               \expandafter\zref@titleref@setcurrent
1937               \expandafter{\ch@pt@c}%
1938             \fi

```



```

1939         \fi
1940     \fi
1941 }%
1942 \ZREF@org@chapter[#{1}]{#2}%
1943 }%
1944 \ZREF@patch{M@sect}{%
1945     \def\M@sect#1#2#3#4#5#6[#7][#8]{%
1946         \ifheadnameref
1947             \zref@titleref@setcurrent{#8}%
1948         \else
1949             \zref@titleref@setcurrent{#7}%
1950         \fi
1951         \ZREF@org@M@sect{#1}{#2}{#3}{#4}{#5}{#6}[#{7}][#{8}]%
1952     }%
1953 }%
1954 }%
1955 }{}%

```

6.15.6 Class beamer

```

1956 \@ifclassloaded{beamer}{%
1957     \ZREF@patch{beamer@section}{%
1958         \long\def\beamer@section[#1]{%
1959             \zref@titleref@setcurrent{#1}%
1960             \ZREF@org@beamer@section[#{1}]%
1961         }%
1962     }%
1963     \ZREF@patch{beamer@subsection}{%
1964         \long\def\beamer@subsection[#1]{%
1965             \zref@titleref@setcurrent{#1}%
1966             \ZREF@org@beamer@subsection[#{1}]%
1967         }%
1968     }%
1969     \ZREF@patch{beamer@subsubsection}{%
1970         \long\def\beamer@subsubsection[#1]{%
1971             \zref@titleref@setcurrent{#1}%
1972             \ZREF@org@beamer@subsubsection[#{1}]%
1973         }%
1974     }%
1975 }{}%

```

6.15.7 Package titlesec

```

1976 \@ifpackageloaded{titlesec}{%
1977     \ZREF@patch{ttl@sect@i}{%
1978         \def\ttl@sect@i#1#2[#3]#4{%
1979             \zref@titleref@setcurrent{#4}%
1980             \ZREF@org@ttl@sect@i{#1}{#2}[#{3}]{#4}%
1981         }%
1982     }%
1983     \ZREF@patch{ttl@straight@i}{%
1984         \def\ttl@straight@i#1[#2]#3{%
1985             \def\ZREF@temp{#2}%
1986             \ifx\ZREF@temp\ltx@empty
1987                 \zref@titleref@setcurrent{#3}%
1988             \else
1989                 \zref@titleref@setcurrent{#2}%
1990             \fi
1991             \ZREF@org@ttl@straight@i{#1}[#{2}]{#3}%
1992         }%

```

```

1993   }%
1994   }{}%

```

6.15.8 Package longtable

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

```

1995   \@ifpackageloaded{longtable}{%
1996     \ZREF@patch{LT@c@ption}{%
1997       \def\LT@c@ption#1[#2]#3{%
1998         \ZREF@org@LT@c@ption{#1}[{#2}]{#3}%
1999         \zref@titleref@setcurrent{#2}%
2000       }%
2001     }%
2002   }{}%

```

6.15.9 Package listings

Package listings: support for its caption.

```

2003   \@ifpackageloaded{listings}{%
2004     \ZREF@patch{lst@MakeCaption}{%
2005       \def\lst@MakeCaption{%
2006         \ifx\lst@label\ltx@empty
2007         \else
2008           \expandafter\zref@titleref@setcurrent\expandafter{%
2009             \lst@caption
2010           }%
2011         \fi
2012         \ZREF@org@lst@MakeCaption
2013       }%
2014     }%
2015   }{}%

```

6.15.10 Theorems

```

2016   \ZREF@patch{@opargbegintheorem}{%
2017     \def\@opargbegintheorem#1#2#3{%
2018       \zref@titleref@setcurrent{#3}%
2019       \ZREF@org@@opargbegintheorem{#1}{#2}{#3}%
2020     }%
2021   }%
2022   \@ifpackageloaded{amsthm}{%
2023     \begingroup
2024       \edef\x{macro:\string#1\string#2[\string#3]}%
2025       \@onelevel@sanitize\x
2026       \def\y#1->#2\@nil{#1}%
2027       \edef\z{\expandafter\y\meaning\@begintheorem->\@nil}%
2028       \@onelevel@sanitize\z
2029       \expandafter\endgroup
2030     \ifx\x\z
2031       \ZREF@patch{@begintheorem}{%
2032         \def\@begintheorem#1#2[#3]{%
2033           \zref@titleref@setcurrent{#3}%
2034           \ZREF@org@@begintheorem{#1}{#2}[{#3}]%
2035         }%
2036       }%
2037     \fi
2038   }{}%

```

```

2039 }
2040 </titleref>
6.16 Module xr
2041 (*xr)
2042 \NeedsTeXFormat{LaTeX2e}
2043 \ProvidesPackage{zref-xr}%
2044 [2020-07-03 v2.32 Module xr for zref (HO)]%
2045 \RequirePackage{zref-base}[2019/11/29]
2046 \ifx\ZREF@base@ok Y%
2047 \else
2048 \expandafter\endinput
2049 \fi

2050 \RequirePackage{keyval}
2051 \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.

```

2052 \zref@newprop{url}{}%
2053 \zref@newprop{urluse}{}%
2054 \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.

```

2055 \providecommand*{\zref@xr@ext}{%
2056 \ltx@ifundefined{XR@ext}{pdf}{\XR@ext}%
2057 }%

```

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

```

2058 \newif\ifZREF@xr@zreflabel

2059 \SetupKeyvalOptions{%
2060 family=ZREF@XR,%
2061 prefix=ZREF@xr%
2062 }
2063 \DeclareBoolOption[true]{tozreflabel}
2064 \DeclareBoolOption[false]{toltxlabel}
2065 \DeclareBoolOption{verbose}
2066 \define@key{ZREF@XR}{ext}{%
2067 \def\zref@xr@{#1}%
2068 }
2069 \DeclareBoolOption[false]{urluse}

```

`\zxrsetup`

```

2070 \newcommand*{\zxrsetup}{%
2071 \kvsetkeys{ZREF@XR}%
2072 }%

```

`\ZREF@xr@URL`

```

2073 \newcount\ZREF@xr@URL
2074 \ZREF@xr@URL=\ltx@zero

```

`\ZREF@xr@AddURL`

```
2075 \def\ZREF@xr@AddURL#1{%
2076   \begingroup
2077   \def\ZREF@temp{#1}%
2078   \count@=\ltx@one
2079   \ZREF@xr@@AddUrl
2080   \endgroup
2081 }
```

`\ZREF@xr@@AddUrl`

```
2082 \def\ZREF@xr@@AddUrl{%
2083   \ifnum\count@>\ZREF@xr@URL
2084     \global\advance\ZREF@xr@URL by\ltx@one
2085     \xdef\ZREF@xr@theURL{\romannumeral\ZREF@xr@URL}%
2086     \global\expandafter\let
2087       \csname Z@U@\ZREF@xr@theURL\endcsname\ZREF@temp
2088     \@PackageInfo{zref-xr}{%
2089       \ltx@backslashchar Z@U@\ZREF@xr@theURL:\MessageBreak
2090       \ZREF@temp\MessageBreak
2091     }%
2092   \else
2093     \expandafter
2094     \ifx\csname Z@U@\romannumeral\count@\endcsname\ZREF@temp
2095       \xdef\ZREF@xr@theURL{\romannumeral\count@}%
2096     \else
2097       \expandafter\expandafter\expandafter\ZREF@xr@@AddUrl
2098     \fi
2099   \fi
2100 }
```

`\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).

```
2101 \ZREF@IfDefinable\zexternaldocument\def{%
2102   {%
2103     \ZREF@NewPropAnchor
2104     \ZREF@NewPropTitle
2105     \begingroup
2106     \csname @safe@actives@true\endcsname
2107     \makeatletter
2108     \@ifstar{%
2109       \ZREF@xr@zreflabelfalse
2110       \@testopt\ZREF@xr@externaldocument{}%
2111     }{%
2112       \ZREF@xr@zreflabeltrue
2113       \@testopt\ZREF@xr@externaldocument{}%
2114     }%
2115   }%
2116 }
```

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.

```
2117 \def\ZREF@xr@externaldocument[#1]#2{%
```

```

2118 \def\ZREF@xr@prefix{#1}%
2119 \let\ZREF@xr@filelist\ltx@empty
2120 \edef\ZREF@xr@externalfile{#2}%
2121 \edef\ZREF@xr@file{\ZREF@xr@externalfile.aux}%
2122 \filename@parse{#2}%
2123 \@testopt\ZREF@xr@graburl{#2.\zref@xr@ext}%
2124 }%
2125 \def\ZREF@xr@graburl[#1]{%
2126 \edef\ZREF@xr@url{#1}%
2127 \ifZREF@xr@urluse
2128 \expandafter\ZREF@xr@AddURL\expandafter{\ZREF@xr@url}%
2129 \expandafter\def\expandafter\ZREF@xr@url
2130 \expandafter{\csname Z@U@\ZREF@xr@theURL\endcsname}%
2131 \fi
2132 \ZREF@xr@checkfile
2133 \endgroup
2134 }%

```

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

```

2135 \def\ZREF@xr@checkfile{%
2136 \openin\@inputcheck\ZREF@xr@file\relax
2137 \ifeof\@inputcheck
2138 \PackageWarning{zref-xr}{%
2139 File '\ZREF@xr@file' not found or empty,\MessageBreak
2140 labels not imported%
2141 }%
2142 \else
2143 \PackageInfo{zref-xr}{%
2144 Label \ifZREF@xr@zreflabel (zref) \fi
2145 import from '\ZREF@xr@file'%
2146 }%
2147 \def\ZREF@xr@found{0}%
2148 \def\ZREF@xr@ignored@empty{0}%
2149 \def\ZREF@xr@ignored@zref{0}%
2150 \def\ZREF@xr@ignored@ltx{0}%
2151 \ZREF@xr@processfile
2152 \closein\@inputcheck
2153 \beginingroup
2154 \let\on@line\ltx@empty
2155 \PackageInfo{zref-xr}{%
2156 Statistics for '\ZREF@xr@file':\MessageBreak
2157 \ZREF@xr@found\space
2158 \ifZREF@xr@zreflabel zref\else LaTeX\fi\space
2159 label(s) found%
2160 \ifnum\ZREF@xr@ignored@empty>0 %
2161 ,\MessageBreak
2162 \ZREF@xr@ignored@empty\space empty label(s) ignored%
2163 \fi
2164 \ifnum\ZREF@xr@ignored@zref>0 %
2165 ,\MessageBreak
2166 \ZREF@xr@ignored@zref\space
2167 duplicated zref label(s) ignored%
2168 \fi
2169 \ifnum\ZREF@xr@ignored@ltx>0 %
2170 ,\MessageBreak
2171 \ZREF@xr@ignored@ltx\space
2172 duplicated latex label(s) ignored%

```

```

2173     \fi
2174   }%
2175   \endgroup
2176 \fi
2177 \ifx\ZREF@xr@filelist\ltx@empty
2178 \else
2179   \edef\ZREF@xr@file{%
2180     \expandafter\ltx@car\ZREF@xr@filelist\@nil
2181   }%
2182   \edef\ZREF@xr@filelist{%
2183     \expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil
2184   }%
2185   \expandafter\ZREF@xr@checkfile
2186 \fi
2187 }%

```

\ZREF@xr@processfile

```

2188 \def\ZREF@xr@processfile{%
2189   \read\@inputcheck to\ZREF@xr@line
2190   \expandafter\ZREF@xr@processline\ZREF@xr@line..\ZREF@nil
2191   \ifeof\@inputcheck
2192   \else
2193     \expandafter\ZREF@xr@processfile
2194   \fi
2195 }%

```

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

```

2196 \long\def\ZREF@xr@processline#1#2#3\ZREF@nil{%
2197   \def\x{#1}%
2198   \toks@{#2}%
2199   \ifZREF@xr@zreflabel
2200     \ifx\x\ZREF@xr@zref@newlabel
2201       \expandafter
2202         \ZREF@xr@process@zreflabel\ZREF@xr@line...\ZREF@nil
2203     \fi
2204   \else
2205     \ifx\x\ZREF@xr@newlabel
2206       \expandafter
2207         \ZREF@xr@process@label\ZREF@xr@line... []\ZREF@nil
2208     \fi
2209   \fi
2210   \ifx\x\ZREF@xr@@input
2211     \edef\ZREF@xr@filelist{%
2212       \etex@unexpanded\expandafter{\ZREF@xr@filelist}%
2213       {\filename@area\the\toks@}%
2214     }%
2215   \fi
2216 }%
2217 \def\ZREF@xr@process@zreflabel\zref@newlabel#1#2#3\ZREF@nil{%
2218   \edef\ZREF@xr@refname{Z@R@\ZREF@xr@prefix#1}%
2219   \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2220   \def\x{#2}%
2221   \edef\ZREF@xr@tempname{$temp$}%
2222   \edef\ZREF@xr@temprefname{Z@R@\ZREF@xr@tempname}%
2223   \let\ZREF@xr@list\x
2224   \ifx\ZREF@xr@list\ltx@empty
2225     \PackageWarningNoLine{zref-xr}{%
2226       Label '#1' without properties ignored\MessageBreak

```

```

2227     in file '\ZREF@xr@file'%
2228 }%
2229 \edef\ZREF@xr@ignored@empty{%
2230   \the\numexpr\ZREF@xr@ignored@empty+1\relax
2231 }%
2232 \else
2233   \expandafter\ZREF@xr@checklist\x\ZREF@nil
2234   \expandafter\let\csname\ZREF@xr@temprefname\endcsname\x
2235   \expandafter\ltx@LocalAppendToMacro
2236   \csname\ZREF@xr@temprefname\expandafter\endcsname
2237   \expandafter{%
2238     \expandafter\externaldocument\expandafter{%
2239       \ZREF@xr@externalfile
2240     }%
2241   }%
2242   \ZREF@xr@urlcheck\ZREF@xr@tempname
2243   \ifZREF@xr@tozreflabel
2244     \@ifundefined{\ZREF@xr@refname}{%
2245       \ifZREF@xr@verbose
2246         \PackageInfo{zref-xr}{%
2247           Import to zref label '\ZREF@xr@tempname#1'%
2248         }%
2249       \fi
2250       \global\expandafter
2251       \let\csname\ZREF@xr@refname\expandafter\endcsname
2252       \csname\ZREF@xr@temprefname\endcsname
2253     }{%
2254       \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2255     }%
2256   \fi
2257   \ifZREF@xr@toltxlabel
2258     \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2259   \fi
2260 \fi
2261 }%
2262 \def\ZREF@xr@process@label\newlabel#1#2#3[#4]#5\ZREF@nil{%
2263   \def\ZREF@xr@refname{Z@R@\ZREF@xr@prefix#1}%
2264   \edef\ZREF@xr@found{\the\numexpr\ZREF@xr@found+1\relax}%
2265   \def\x{#2}%
2266   \edef\ZREF@xr@tempname{\$temp$}%
2267   \edef\ZREF@xr@temprefname{Z@R@\ZREF@xr@tempname}%
2268   \expandafter\ZREF@xr@scanparams
2269     \csname\ZREF@xr@temprefname\expandafter\endcsname
2270     \x}{-}{-}{-}\ZREF@nil
2271 \ifx\#4\%
2272 \else
2273   % ntheorem knows an optional argument at the end of \newlabel
2274   \ZREF@NewPropTheotype
2275   \expandafter\ltx@LocalAppendToMacro
2276     \csname\ZREF@xr@temprefname\endcsname{\thetype{#4}}%
2277 \fi
2278 \expandafter\ltx@LocalAppendToMacro
2279 \csname\ZREF@xr@temprefname\expandafter\endcsname\expandafter{%
2280   \expandafter\externaldocument\expandafter{%
2281     \ZREF@xr@externalfile
2282   }%
2283 }%
2284 \ZREF@xr@urlcheck\ZREF@xr@tempname

```

```

2285 \ifZREF@xr@tozreflabel
2286   \@ifundefined{\ZREF@xr@refname}{%
2287     \ifZREF@xr@verbose
2288       \PackageInfo{zref-xr}{%
2289         Import to zref label ‘\ZREF@xr@prefix#1’%
2290       }%
2291     \fi
2292     \global\expandafter
2293     \let\csname\ZREF@xr@refname\expandafter\endcsname
2294     \csname\ZREF@xr@temprefname\endcsname
2295   }{%
2296     \ZREF@xr@zref@ignorewarning{\ZREF@xr@prefix#1}%
2297   }%
2298 \fi
2299 \ifZREF@xr@toltxlabel
2300   \ZREF@xr@tolabel{\ZREF@xr@tempname}{\ZREF@xr@prefix#1}%
2301 \fi
2302 }
2303 \def\ZREF@xr@zref@newlabel{\zref@newlabel}%
2304 \def\ZREF@xr@newlabel{\newlabel}%
2305 \def\ZREF@xr@input{\@input}%
2306 \def\ZREF@xr@relax{\relax}%

```

\ZREF@xr@tolabel

```

2307 \def\ZREF@xr@tolabel#1#2{%
2308   \ifZREF@xr@verbose
2309     \PackageInfo{zref-xr}{%
2310       Import to LaTeX label ‘#2’%
2311     }%
2312   \fi
2313   \zref@wrapper@unexpanded{%
2314     \expandafter\xdef\csname r@#2\endcsname{%
2315       {%
2316         \ltx@ifundefined{M@TitleReference}{%
2317           \ltx@ifundefined{TR@TitleReference}{%
2318             \zref@extractdefault{#1}{default}{}%
2319           }{%
2320             \noexpand\TR@TitleReference
2321             {\zref@extractdefault{#1}{default}{}}%
2322             {\zref@extractdefault{#1}{title}{}}%
2323           }%
2324         }{%
2325           \noexpand\M@TitleReference
2326           {\zref@extractdefault{#1}{default}{}}%
2327           {\zref@extractdefault{#1}{title}{}}%
2328         }%
2329       }%
2330     {\zref@extractdefault{#1}{page}{}}%
2331     \ltx@ifpackageloaded{nameref}{%
2332       {\zref@extractdefault{#1}{title}{}}%
2333       {\zref@extractdefault{#1}{anchor}{}}%
2334       \zref@ifrefcontainsprop{#1}{urluse}{%
2335         {\zref@extractdefault{#1}{urluse}{}}%
2336       }{%
2337         {\zref@extractdefault{#1}{url}{}}%
2338       }%
2339     }{}%
2340   }%

```



```
2341 }%
2342 }
```

`\ZREF@xr@zref@ignorewarning`

```
2343 \def\ZREF@xr@zref@ignorewarning#1{%
2344 \PackageWarningNoLine{zref-xr}{%
2345   Zref label '#1' is already in use\MessageBreak
2346   in file '\ZREF@xr@file'%
2347 }%
2348 \edef\ZREF@xr@ignored@zref{%
2349   \the\numexpr\ZREF@xr@ignored@zref+1%
2350 }%
2351 }%
```

`\ZREF@xr@ltx@ignorewarning`

```
2352 \def\ZREF@xr@ltx@ignorewarning#1{%
2353 \PackageWarningNoLine{zref-xr}{%
2354   LaTeX label '#1' is already in use\MessageBreak
2355   in file '\ZREF@xr@file'%
2356 }%
2357 \edef\ZREF@xr@ignored@ltx{%
2358   \the\numexpr\ZREF@xr@ignored@ltx+1%
2359 }%
2360 }%
```

`\ZREF@xr@checklist`

```
2361 \def\ZREF@xr@checklist#1#2#3\ZREF@nil{%
2362 \ifx\@undefined#1\relax
2363 \expandafter\ZREF@xr@checkkey\string#1\@nil
2364 \fi
2365 \ifx\#3\%
2366 \else
2367 \ltx@ReturnAfterFi{%
2368 \ZREF@xr@checklist#3\ZREF@nil
2369 }%
2370 \fi
2371 }%
2372 \def\ZREF@xr@checkkey#1#2\@nil{%
2373 \zref@ifpropundefined{#2}{%
2374 \zref@newprop{#2}{}%
2375 }{}%
2376 }%
```

`\ZREF@xr@scanparams`

```
2377 \def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{%
2378 \let#1\ltx@empty
2379 \ZREF@foundfalse
2380 \ZREF@xr@scantitleref#1#2\TR@TitleReference{}{} \ZREF@nil
2381 \ifZREF@found
2382 \else
2383 \ltx@LocalAppendToMacro#1{\default{#2}}%
2384 \fi
2385 % page
2386 \ltx@LocalAppendToMacro#1{\page{#3}}%
2387 % nameref title
2388 \ifZREF@found
2389 \else
2390 \ifx\#4\%
```

```

2391 \else
2392 \def\ZREF@xr@temp{#4}%
2393 \ifx\ZREF@xr@temp\ZREF@xr@relax
2394 \else
2395 \ltx@LocalAppendToMacro#1{\title{#4}}%
2396 \fi
2397 \fi
2398 \fi
2399 % anchor
2400 \ifx\#5\%
2401 \else
2402 \ltx@LocalAppendToMacro#1{\anchor{#5}}%
2403 \fi
2404 \ifx\#6\%
2405 \else
2406 \ifZREF@xr@urluse
2407 \ZREF@xr@AddURL{#6}%
2408 \expandafter\ltx@LocalAppendToMacro\expandafter#1%
2409 \expandafter{%
2410 \expandafter\urluse\expandafter{%
2411 \csname Z@U@\ZREF@xr@theURL\endcsname
2412 }%
2413 }%
2414 \else
2415 \ltx@LocalAppendToMacro#1{\url{#6}}%
2416 \fi
2417 \fi
2418 }%

```

\ZREF@xr@scantitleref

```

2419 \def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{%
2420 \ifx\#5\%
2421 \else
2422 \ltx@LocalAppendToMacro#1{%
2423 \default{#3}%
2424 \title{#4}%
2425 }%
2426 \ZREF@foundtrue
2427 \fi
2428 }%

```

\ZREF@xr@urlcheck

```

2429 \def\ZREF@xr@urlcheck#1{%
2430 \zref@ifrefcontainsprop{#1}{anchor}{%
2431 \zref@ifrefcontainsprop{#1}{url}{%
2432 }{%
2433 \expandafter
2434 \ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
2435 \expandafter{%
2436 \csname url\ifZREF@xr@urluse use\fi
2437 \expandafter\endcsname\expandafter{\ZREF@xr@url}%
2438 }%
2439 }%
2440 }{%
2441 }%
2442 }%
2443 \</xr>

```

6.17 Module hyperref

```
UNFINISHED :-(
2444 (*hyperref)
2445 \NeedsTeXFormat{LaTeX2e}
2446 \ProvidesPackage{zref-hyperref}%
2447   [2020-07-03 v2.32 Module hyperref for zref (HO)]%
2448 \RequirePackage{zref-base}[2019/11/29]
2449 \ifx\ZREF@base@ok Y%
2450 \else
2451   \expandafter\endinput
2452 \fi

2453 \ZREF@NewPropAnchor
2454 \zref@addprop\ZREF@mainlist{anchor}%
2455 \end{hyperref}
```

6.18 Module savepos

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

6.18.1 Identification

```
2456 (*savepos)
2457 \NeedsTeXFormat{LaTeX2e}
2458 \ProvidesPackage{zref-savepos}%
2459   [2020-07-03 v2.32 Module savepos for zref (HO)]%
2460 \RequirePackage{zref-base}[2019/11/29]
2461 \ifx\ZREF@base@ok Y%
2462 \else
2463   \expandafter\endinput
2464 \fi
```

6.18.2 Availability

First we check, whether the feature is available.

```
2465 \ifx\directlua@undefined
2466 \ltx@ifundefined{pdfsavepos}{%
2467   \PackageError\ZREF@name{%
2468     \string\pdfsavepos\space is not supported.\MessageBreak
2469     It is provided by pdfTeX (1.40) or XeTeX%
2470   }\ZREF@UpdatePdfTeX
2471 \endinput
2472 }{}%
2473 \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`.

```
2474 \ifpdf
2475 \else
2476   \ltx@ifundefined{pdftexversion}{%
2477     }{%
2478       \ifnum\pdftexversion<140 %
2479         \PackageError\ZREF@name{%
2480           \string\pdfsavepos\space is not supported in DVI mode%
2481           \MessageBreak
2482           of this pdfTeX version%

```

```

2483     }\ZREF@UpdatePdfTeX
2484     \expandafter\expandafter\expandafter\endinput
2485     \fi
2486   }%
2487 \fi

```

6.18.3 Setup

```

2488 \zref@newlist{savepos}
2489 \ifx\directlua\@undefined
2490   \zref@newprop*{posx}[0]{\the\pdflastxpos}
2491   \zref@newprop*{posy}[0]{\the\pdflastypos}
2492 \else
2493   \zref@newprop*{posx}[0]{\the\lastxpos}
2494   \zref@newprop*{posy}[0]{\the\lastypos}
2495 \fi
2496 \zref@addprops{savepos}{posx,posy}

```

6.18.4 User macros

`\zref@savepos`

```

2497 \ifx\directlua\@undefined
2498   \def\zref@savepos{%
2499     \if@filesw
2500       \pdfsavepos
2501     \fi
2502   }
2503 \else
2504   \def\zref@savepos{%
2505     \if@filesw
2506       \savepos
2507     \fi
2508   }
2509 \fi

```

`\ZREF@zsavepos`

```

2510 \def\ZREF@zsavepos#1#2#3{%
2511   \@bsphack
2512   \if@filesw
2513     \zref@savepos
2514     #1{#3}{#2}%
2515     \ltx@ifundefined{TeXeTstate}{%
2516       }{%
2517         \ifnum\TeXeTstate=\ltx@zero
2518         \else
2519           \zref@savepos
2520         \fi
2521       }%
2522     \fi
2523   \@esphack
2524 }

```

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

```

2525 \ZREF@ifdefinable\zsavepos\def{%
2526   {%
2527     \ZREF@zsavepos\zref@labelbylist{savepos}%
2528   }%
2529 }

```

```

\zsaveposx
2530 \ZREF@ifDefinable\zsaveposx\def{%
2531   {%
2532     \ZREF@zsavepos\zref@labelbyprops{posx}%
2533   }%
2534 }

```

```

\zsaveposy
2535 \ZREF@ifDefinable\zsaveposy\def{%
2536   {%
2537     \ZREF@zsavepos\zref@labelbyprops{posy}%
2538   }%
2539 }

```

`\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.

```

2540 \newcommand*{\zposx}[1]{%
2541   \zref@extract{#1}{posx}%
2542 }%
2543 \newcommand*{\zposy}[1]{%
2544   \zref@extract{#1}{posy}%
2545 }%

```

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

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

```

\ZREF@savepos@ok
2546 \let\ZREF@savepos@ok=Y
2547 
```

6.19 Module `abspos`

6.19.1 Identification

```

2548 (*abspos)
2549 \NeedsTeXFormat{LaTeX2e}
2550 \ProvidesPackage{zref-abspos}%
2551   [2020-07-03 v2.32 Module abspos for zref (HO)]%
2552 \RequirePackage{zref-base}[2019/11/29]
2553 \ifx\ZREF@base@ok Y%
2554 \else
2555   \expandafter\endinput
2556 \fi

2557 \RequirePackage{zref-savepos}[2019/11/29]
2558 \ifx\ZREF@savepos@ok Y%
2559 \else
2560   \expandafter\endinput
2561 \fi

2562 \RequirePackage{zref-pagelayout}[2019/11/29]
2563 \zref@addprop{savepos}{abspage}
2564 \zref@addprop{savepos}{pagevalue}

```

\zref@absposx

```
2565 \newcommand*{\zref@absposx}[3]{%
2566   \number
2567   \expandafter\zref@absposnumx\expandafter{%
2568     \number\zref@extractdefault{#1}{abspage}{0}%
2569   }{#2}{#3}%
2570   \ltx@space
2571 }
```

\zref@absposy

```
2572 \newcommand*{\zref@absposy}[3]{%
2573   \number
2574   \expandafter\zref@absposnumy\expandafter{%
2575     \number\zref@extractdefault{#1}{abspage}{0}%
2576   }{#2}{#3}%
2577   \ltx@space
2578 }
```

\zref@absposnumx

```
2579 \newcommand*{\zref@absposnumx}[3]{%
2580   \number
2581   % \ifnum#1>\ltx@zero
2582   % \zref@ifrefundefined{thepage#1}{%
2583   %   0%
2584   % }{%
2585   % \numexpr\ZREF@absposnum{thepage#1}{#2}{x}{#3}\relax
2586   % }%
2587   % \else
2588   %   0%
2589   % \fi
2590 }
```

\zref@absposnumy

```
2591 \newcommand*{\zref@absposnumy}[3]{%
2592   \number
2593   % \ifnum#1>\ltx@zero
2594   % \zref@ifrefundefined{thepage#1}{%
2595   %   0%
2596   % }{%
2597   % \numexpr\ZREF@absposnum{thepage#1}{#2}{y}{#3}\relax
2598   % }%
2599   % \else
2600   %   0%
2601   % \fi
2602 }
```

\ZREF@absposnum

```
2603 \def\ZREF@absposnum#1#2#3#4{%
2604   \ltx@ifundefined{ZREF@abspos@#2@#3@#4}{%
2605     0%
2606   }{%
2607     \csname ZREF@abspos@#2@#3@#4\endcsname{#1}%
2608   }%
2609 }
```

\zref@def@absposx

```
2610 \ZREF@Robust\def\zref@def@absposx#1{%
```

```

2611 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposx}%
2612 }

\zref@def@absposy
2613 \ZREF@Robust\def\zref@def@absposy#1{%
2614 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
2615 }

\zref@def@absposnumx
2616 \ZREF@Robust\def\zref@def@absposnumx#1{%
2617 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumx}%
2618 }

\zref@def@absposnumy
2619 \ZREF@Robust\def\zref@def@absposnumy#1{%
2620 \zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposnumy}%
2621 }

\ZREF@def@abspos
2622 \def\ZREF@def@absposnumy#1#2#3#4#5{%
2623 \edef#1{#2{#3}{#4}{#5}}%
2624 }

\zref@absposused
2625 \ZREF@Robust\def\zref@absposused{%
2626 \zref@wrapper@babel\ZREF@abspos@used
2627 }

\ZREF@abspos@used
2628 \def\ZREF@abspos@used#1{%
2629 \zref@refused{#1}%
2630 \zref@ifrefundefined{#1}{%
2631 }{%
2632 \begingroup
2633 \edef\ZREF@temp{%
2634 \zref@extractdefault{#1}{abspage}{0}%
2635 }%
2636 \ifnum\ZREF@temp>\ltx@zero
2637 \zref@refused{thepage\ZREF@temp}%
2638 \else
2639 \@PackageError{zref-abspos}{%
2640 \string\zref@pos@label@used\ltx@space
2641 needs property 'abspage'\MessageBreak
2642 in label '#1'%
2643 }\@ehc
2644 \fi
2645 \endgroup
2646 }%
2647 }

\zref@absposnumused
2648 \newcommand*{\zref@absposnumused}[1]{%
2649 \ifnum#1>\ltx@zero
2650 \zref@refused{thepage\number#1}%
2651 \else
2652 \@PackageError{zref-abspos}{%
2653 Invalid absolute page number (#1)\MessageBreak

```

```

2654     for \string\zref@pos@num@used.\MessageBreak
2655     A positive integer is expected%
2656   }\@ehc
2657 \fi
2658 }

```

\zref@ifabsposundefined

```

2659 \def\zref@ifabsposundefined#1{%
2660   \zref@ifrefundefined{#1}\ltx@firsttwo{%
2661     \expandafter\zref@ifabsposnumundefined\expandafter{%
2662       \number\zref@extractdefault{#1}{abspage}{0}%
2663     }%
2664   }%
2665 }

```

\zref@ifabsposnumundefined

```

2666 \def\zref@ifabsposnumundefined#1{%
2667   \ifnum\ZREF@number{#1}>\ltx@zero
2668     \zref@ifrefundefined{thepage#1}%
2669     \ltx@firstoftwo\ltx@secondoftwo
2670   \else
2671     \expandafter\ltx@firstoftwo
2672   \fi
2673 }

```

6.19.2 Media

\ZREF@abspos@media@width

```

2674 \edef\ZREF@abspos@media@width{%
2675   \ltx@ifundefined{pdfpagewidth}{%
2676     \ltx@ifundefined{mediawidth}{%
2677       \ltx@ifundefined{stockwidth}{%
2678         paperwidth%
2679       }{%
2680         stockwidth%
2681       }%
2682     }{%
2683       mediawidth%
2684     }%
2685   }{%
2686     pdfpagewidth%
2687   }%
2688 }
2689 \ifluatex
2690 \def\ZREF@abspos@media@width{pdfpagewidth}%
2691 \fi

```

\ZREF@abspos@media@height

```

2692 \edef\ZREF@abspos@media@height{%
2693   \ltx@ifundefined{pdfpageheight}{%
2694     \ltx@ifundefined{mediaheight}{%
2695       \ltx@ifundefined{stockheight}{%
2696         paperheight%
2697       }{%
2698         stockheight%
2699       }%
2700     }{%
2701       mediaheight%

```



```

2702   }%
2703 }{%
2704   \noexpand\ifcase\pdfpageheight
2705   \ltx@ifundefined{stockheight}{%
2706     paperheight%
2707   }{%
2708     stockheight%
2709   }%
2710   \noexpand\else
2711     pdfpageheight%
2712   \noexpand\fi
2713 }%
2714 }
2715 \ifluatex
2716 \edef\ZREF@abspos@media@height{%
2717   \noexpand\ifcase\pageheight
2718   \ltx@ifundefined{stockheight}{%
2719     paperheight%
2720   }{%
2721     stockheight%
2722   }%
2723   \noexpand\else
2724     pdfpageheight%
2725   \noexpand\fi}%
2726 \fi

\ZREF@abspos@media@x@left
2727 \def\ZREF@abspos@media@x@left#1{%
2728   0%
2729 }

\ZREF@abspos@media@x@right
2730 \def\ZREF@abspos@media@x@right#1{%
2731   \zref@extract{#1}\ZREF@abspos@media@width
2732 }

\ZREF@abspos@media@x@center
2733 \def\ZREF@abspos@media@x@center#1{%
2734   \ZREF@abspos@media@x@left{#1}%
2735   +\zref@extract{#1}\ZREF@abspos@media@width/2%
2736 }

\ZREF@abspos@media@y@top
2737 \def\ZREF@abspos@media@y@top#1{%
2738   \zref@extract{#1}\ZREF@abspos@media@height
2739 }

\ZREF@abspos@media@y@bottom
2740 \def\ZREF@abspos@media@y@bottom#1{%
2741   0%
2742 }

\ZREF@abspos@media@y@center
2743 \def\ZREF@abspos@media@y@center#1{%
2744   \zref@extract{#1}\ZREF@abspos@media@height/2%
2745 }

```

6.19.3 Paper

```
\ZREF@abspos@paper@x@left
2746 \def\ZREF@abspos@paper@x@left#1{%
2747   0%
2748 }

\ZREF@abspos@paper@x@right
2749 \def\ZREF@abspos@paper@x@right#1{%
2750   \zref@extract{#1}{paperwidth}%
2751 }

\ZREF@abspos@paper@x@center
2752 \def\ZREF@abspos@paper@x@center#1{%
2753   \zref@extract{#1}{paperwidth}/2%
2754 }

\ZREF@abspos@paper@y@top
2755 \let\ZREF@abspos@paper@y@top\ZREF@abspos@media@y@top

\ZREF@abspos@paper@y@bottom
2756 \def\ZREF@abspos@paper@y@bottom#1{%
2757   \ZREF@abspos@paper@y@top{#1}%
2758   -\zref@extract{#1}{paperheight}%
2759 }

\ZREF@abspos@paper@y@center
2760 \def\ZREF@abspos@paper@y@center#1{%
2761   \ZREF@abspos@paper@y@top{#1}%
2762   -\zref@extract{#1}{paperheight}/2%
2763 }
```

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
2764 \let\ZREF@temp\ltx@two
2765 \ltx@ifundefined{pdfhorigin}{}{%
2766 % \ifpdf
2767   \let\ZREF@temp\ltx@zero
2768 % \fi
2769 }
2770 \ifluatex
2771 % \ifpdf
2772   \let\ZREF@temp\ltx@zero
2773 % \fi
2774 \fi
2775
2776 \ifx\ZREF@temp\ltx@two
2777   \ifnum\mag=1000 %
2778     \let\ZREF@temp\ltx@one
2779   \fi
2780 \fi
2781 \ifcase\ZREF@temp
2782   \def\ZREF@abspos@origin@x#1{%
2783     \zref@extract{#1}{pdfhorigin}%
```

```

2784 }%
2785 \or
2786 \def\ZREF@abspos@origin@x#1{%
2787   4736286%
2788 }%
2789 \or
2790 \def\ZREF@abspos@origin@x#1{%
2791   \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2792 }%
2793 \fi

```

\ZREF@abspos@origin@y

```

2794 \let\ZREF@temp\ltx@two
2795 \ltx@ifundefined{pdfvorigin}{}{%
2796 % \ifpdf
2797   \let\ZREF@temp\ltx@zero
2798 % \fi
2799 }
2800 \ifluatex
2801 % \ifpdf
2802   \let\ZREF@temp\ltx@zero
2803 % \fi
2804 \fi
2805 \ifx\ZREF@temp\ltx@two
2806   \ifnum\mag=1000 %
2807     \let\ZREF@temp\ltx@one
2808   \fi
2809 \fi
2810 \ifcase\ZREF@temp
2811   \def\ZREF@abspos@origin@y#1{%
2812     \zref@extract{#1}{pdfvorigin}%
2813   }%
2814 \or
2815   \def\ZREF@abspos@origin@y#1{%
2816     4736286%
2817   }%
2818 \or
2819   \def\ZREF@abspos@origin@y#1{%
2820     \numexpr\mag/1000*\dimexpr 1truein\relax\relax
2821   }%
2822 \fi

```

6.19.5 Header

\ZREF@abspos@head@x@left

```

2823 \def\ZREF@abspos@head@x@left#1{%
2824   \ZREF@abspos@paper@x@left{#1}%
2825   +\ZREF@abspos@origin@x{#1}%
2826   +\zref@extract{#1}{hoffset}%
2827   +\ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2828     \zref@extract{#1}{oddsidemargin}%
2829   \else
2830     \zref@extract{#1}{evensidemargin}%
2831   \fi
2832 }

```

\ZREF@abspos@head@x@right

```

2833 \def\ZREF@abspos@head@x@right#1{%

```

```

2834 \ZREF@abspos@head@x@left{#1}%
2835 +\zref@extract{#1}{textwidth}%
2836 }

```

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

```

2837 \def\ZREF@abspos@head@x@center#1{%
2838 \ZREF@abspos@head@x@left{#1}%
2839 +\zref@extract{#1}{textwidth}/2%
2840 }

```

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

```

2841 \def\ZREF@abspos@head@y@top#1{%
2842 \ZREF@abspos@paper@y@top{#1}%
2843 -\ZREF@abspos@origin@y{#1}%
2844 -\zref@extract{#1}{voffset}%
2845 -\zref@extract{#1}{topmargin}%
2846 }

```

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

```

2847 \def\ZREF@abspos@head@y@bottom#1{%
2848 \ZREF@abspos@head@y@top{#1}%
2849 -\zref@extract{#1}{headheight}%
2850 }

```

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

```

2851 \def\ZREF@abspos@head@y@center#1{%
2852 \ZREF@abspos@head@y@top{#1}%
2853 -\zref@extract{#1}{headheight}/2%
2854 }

```

6.19.6 Body

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

```

2855 \let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left

```

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

```

2856 \let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right

```

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

```

2857 \let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center

```

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

```

2858 \def\ZREF@abspos@body@y@top#1{%
2859 \ZREF@abspos@head@y@bottom{#1}%
2860 -\zref@extract{#1}{headsep}%
2861 }

```

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

```

2862 \def\ZREF@abspos@body@y@bottom#1{%
2863 \ZREF@abspos@body@y@top{#1}%
2864 -\zref@extract{#1}{textheight}%
2865 }

```

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

```

2866 \def\ZREF@abspos@body@y@center#1{%
2867 \ZREF@abspos@body@y@top{#1}%
2868 -\zref@extract{#1}{textheight}/2%
2869 }

```

6.19.7 Footer

```
\ZREF@abspos@foot@x@left
2870 \let\ZREF@abspos@foot@x@left\ZREF@abspos@head@x@left

\ZREF@abspos@foot@x@right
2871 \let\ZREF@abspos@foot@x@right\ZREF@abspos@head@x@right

\ZREF@abspos@foot@x@center
2872 \let\ZREF@abspos@foot@x@center\ZREF@abspos@head@x@center

\ZREF@abspos@foot@y@bottom
2873 \def\ZREF@abspos@foot@y@bottom#1{%
2874   \ZREF@abspos@body@y@bottom{#1}%
2875   -\zref@extract{#1}{footskip}%
2876 }
```

6.19.8 Marginal notes

```
\ZREF@abspos@marginpar@x@left
2877 \def\ZREF@abspos@marginpar@x@left#1{%
2878   \ifodd\zref@extractdefault{#1}{pagevalue}{\number\c@page} %
2879   \ZREF@abspos@body@x@right{#1}%
2880   +\zref@extract{#1}{marginparsep}%
2881   \else
2882   \ZREF@abspos@body@x@left{#1}%
2883   -\zref@extract{#1}{marginparsep}%
2884   -\zref@extract{#1}{marginparwidth}%
2885   \fi
2886 }

\ZREF@abspos@marginpar@x@right
2887 \def\ZREF@abspos@marginpar@x@right#1{%
2888   \ZREF@abspos@marginpar@x@left{#1}%
2889   +\zref@extract{#1}{marginparwidth}%
2890 }

\ZREF@abspos@marginpar@x@center
2891 \def\ZREF@abspos@marginpar@x@center#1{%
2892   \ZREF@abspos@marginpar@x@left{#1}%
2893   +\zref@extract{#1}{marginparwidth}/2%
2894 }

\ZREF@abspos@marginpar@y@top
2895 \let\ZREF@abspos@marginpar@y@top\ZREF@abspos@body@y@top

\ZREF@abspos@marginpar@y@bottom
2896 \let\ZREF@abspos@marginpar@y@bottom\ZREF@abspos@body@y@bottom

\ZREF@abspos@marginpar@y@center
2897 \let\ZREF@abspos@marginpar@y@center\ZREF@abspos@body@y@center
```

6.19.9 Stock paper

```
\ZREF@abspos@stock@x@left
2898 \let\ZREF@abspos@stock@x@left\ZREF@abspos@paper@x@left

\ZREF@abspos@stock@x@right
2899 \let\ZREF@abspos@stock@x@right\ZREF@abspos@paper@x@right

\ZREF@abspos@stock@x@center
2900 \let\ZREF@abspos@stock@x@center\ZREF@abspos@paper@x@center

\ZREF@abspos@stock@y@top
2901 \let\ZREF@abspos@stock@y@top\ZREF@abspos@paper@y@top

\ZREF@abspos@stock@y@bottom
2902 \let\ZREF@abspos@stock@y@bottom\ZREF@abspos@paper@y@bottom

\ZREF@abspos@stock@y@center
2903 \let\ZREF@abspos@stock@y@center\ZREF@abspos@paper@y@center
2904 (/abspos)
```

6.20 Module dotfill

```
2905 (*dotfill)
2906 \NeedsTeXFormat{LaTeX2e}
2907 \ProvidesPackage{zref-dotfill}%
2908 [2020-07-03 v2.32 Module dotfill for zref (HO)]%
2909 \RequirePackage{zref-base}[2019/11/29]
2910 \ifx\ZREF@base@ok Y%
2911 \else
2912 \expandafter\endinput
2913 \fi
```

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

```
2914 \RequirePackage{zref-savepos}[2019/11/29]
```

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

```
2915 \zref@require@unique
```

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

```
2916 \RequirePackage{keyval}
```

The definitions of the keys follow.

```
2917 \define@key{ZREF@DF}{unit}{%
2918 \def\ZREF@df@unit{#1}%
2919 }
2920 \define@key{ZREF@DF}{min}{%
2921 \def\ZREF@df@min{#1}%
2922 }
2923 \define@key{ZREF@DF}{dot}{%
2924 \def\ZREF@df@dot{#1}%
2925 }
```

Defaults are set, see user interface.

```
2926 \providecommand\ZREF@df@min{2}
2927 \providecommand\ZREF@df@unit{.44em}
2928 \providecommand\ZREF@df@dot{.}
```

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

```
2929 \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.

```

2930 \ZREF@ifDefinable\zdotfill\def{%
2931   {%
2932     \leavevmode
2933     \global\advance\c@zref@unique\ltx@one
2934     \begingroup
2935     \def\ZREF@temp{zref@\number\c@zref@unique}%
2936     \pdfsavepos
2937     \zref@labelbyprops{\thezref@unique L}{posx}%
2938     \setlength{\dimen@}{\ZREF@df@unit}%
2939     \zref@ifrefundefined{\thezref@unique R}{%
2940       \ZREF@dotfill
2941     }{%
2942       \ifnum\numexpr\zposx{\thezref@unique R}%
2943         -\zposx{\thezref@unique L}\relax
2944         <\dimexpr\ZREF@df@min\dimen@\relax
2945         \hfill
2946       \else
2947         \ZREF@dotfill
2948       \fi
2949     }%
2950     \pdfsavepos
2951     \zref@labelbyprops{\thezref@unique R}{posx}%
2952   \endgroup
2953   \kern\z@
2954 }%
2955 }

```

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

```

2956 \def\ZREF@dotfill{%
2957   \cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill
2958 }

2959 </dotfill>

```

6.21 Module `env`

```

2960 (*env)
2961 \NeedsTeXFormat{LaTeX2e}
2962 \ProvidesPackage{zref-env}%
2963   [2020-07-03 v2.32 Module env for zref (HO)]%
2964 \RequirePackage{zref-base}[2019/11/29]
2965 \ifx\ZREF@base@ok Y%
2966 \else
2967   \expandafter\endinput
2968 \fi

2969 \zref@newprop{envname}[]{\@currenvir}
2970 \zref@newprop{envline}[]{\zref@env@line}

```

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

```

2971 \def\zref@env@line{%
2972   \ifx\@currenvline\ltx@empty
2973   \else
2974     \expandafter
2975     \ZREF@ENV@line\@currenvline\ltx@empty line \ltx@empty\@nil
2976   \fi
2977 }

```

\ZREF@ENV@line

```
2978 \def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}%  
2979 \end{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](#)

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

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

7.4 Refresh file name databases

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

7.5 Some details for the interested

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

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

\LaTeX : Generate the documentation.

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

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

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

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

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

An example follows how to generate the documentation with `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 \LaTeX .

[2007/04/06 v1.6]

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

[2007/04/17 v1.7]

- Package `atbegshi` replaces `everyshi`.

[2007/04/22 v1.8]

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

[2007/05/02 v1.9]

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

[2007/05/06 v2.0]

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

[2007/05/28 v2.1]

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

[2008/09/21 v2.2]

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

[2008/10/01 v2.3]

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

[2009/08/07 v2.4]

- Module `runs` added.

[2009/12/06 v2.5]

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

[2009/12/07 v2.6]

- Version date added for package `atveryend`.

[2009/12/08 v2.7]

- Module `titleref`: Use of package `getttitlestring`.

[2010/03/26 v2.8]

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

[2010/03/29 v2.9]

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

[2010/04/08 v2.10]

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

[2010/04/15 v2.11]

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

[2010/04/17 v2.12]

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

[2010/04/19 v2.13]

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

[2010/04/22 v2.14]

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

[2010/04/23 v2.15]

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

[2010/04/28 v2.16]

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

[2010/05/01 v2.17]

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

[2010/05/13 v2.18]

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

[2010/10/22 v2.19]

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

[2011/02/12 v2.20]

- Fix for warning in `zref-xr`.

[2011/03/18 v2.21]

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

[2011/10/05 v2.22]

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

[2011/12/05 v2.23]

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

[2012/04/04 v2.24]

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

[2016/05/16 v2.25]

- Documentation updates.

[2016/05/21 v2.26]

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

[2018/11/21 v2.27]

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

[2019/11/29 v2.28]

- Documentation updates.
- Use iftex directly.

[2020-03-03 v2.29]

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

[2020-03-04 v2.30]

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

[2020-05-28 v2.31]

- Adapted module zref-counter to use `\@currentcounter` in the next L^AT_EXversion (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.

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>\@auxout</code> | 716 |
| <code>\@PackageError</code> | <code>\@begintheorem</code> | 2027, 2032 |
| <code>\@PackageInfo</code> | <code>\@bsphack</code> | 601, 611, 631, 2511 |
| <code>\@PackageInfoNoLine</code> | <code>\@caption</code> | 1872 |
| 549, 564, 1352, 1447, 1459, 1527, 1579 | <code>\@chapter</code> | 1884, 1921 |
| <code>\@PackageWarning</code> | <code>\@currentHref</code> | 941 |
| <code>\@addtoreset</code> | <code>\@currentcounter</code> | 1030 |
| 912, 1009 | | |

| | | | |
|--------------------------------------|--|---------------------------------------|---|
| <code>\@currentlabel</code> | 936 | <code>\AtBeginShipoutBoxDepth</code> | 1323 |
| <code>\@currenvir</code> | 2969 | <code>\AtBeginShipoutBoxHeight</code> | 1322 |
| <code>\@currrentline</code> | 2972, 2975 | <code>\AtBeginShipoutBoxWidth</code> | 1321 |
| <code>\@ehc</code> | 297, 307, 492, 515, 527, 1611, 2643, 2656 | <code>\AtEndOfPackage</code> | 195 |
| <code>\@esphack</code> | 608, 628, 643, 2523 | <code>\AtVeryEndDocument</code> | 1343, 1504 |
| <code>\@firstofone</code> | 1567 | B | |
| <code>\@firstoftwo</code> | 1432, 1511 | <code>\beamer@section</code> | 1958 |
| <code>\@ifclassloaded</code> | 1919, 1956 | <code>\beamer@subsection</code> | 1964 |
| <code>\@ifdefinable</code> | 243, 290 | <code>\beamer@subsubsection</code> | 1970 |
| <code>\@ifl@t@r</code> | 223, 1028 | <code>\begin</code> | 23, 57, 100, 106, 156, 172 |
| <code>\@ifnextchar</code> | 531, 1738 | <code>\bfseries</code> | 929 |
| <code>\@ifpackageloaded</code> | 1976, 1995, 2003, 2022 | C | |
| <code>\@ifstar</code> | 496, 2108 | <code>\c@abspage</code> | 1016, 1770, 1777 |
| <code>\@ifundefined</code> | 192, 910, 1749, 2244, 2286 | <code>\c@page</code> | 955, 1061, 1767, 2827, 2878 |
| <code>\@input</code> | 2305 | <code>\c@zpage</code> | 1767, 1773 |
| <code>\@inputcheck</code> | 2136, 2137, 2152, 2189, 2191 | <code>\c@zref@unique</code> | 916, 1762, 2933, 2935 |
| <code>\@latex@warning</code> | 778 | <code>\ch@pt@c</code> | 1925, 1931, 1937 |
| <code>\@mainaux</code> | 1696 | <code>\chapter</code> | 24, 30, 32, 61, 82 |
| <code>\@namedef</code> | 536 | <code>\ChapterPages</code> | 91, 112 |
| <code>\@newl@bel</code> | 286 | <code>\ChapterStart</code> | 78, 135, 150, 166 |
| <code>\@nil</code> | 1626, 1628, 1837, 1844, 2026, 2027, 2180, 2183, 2363, 2372, 2975, 2978 | <code>\ChapterStop</code> | 85, 148, 165, 184 |
| <code>\@onelevel@sanitize</code> | 423, 441, 506, 534, 2025, 2028 | <code>\chardef</code> | 1167, 1182, 1191, 1195 |
| <code>\@opargbegintheorem</code> | 2017 | <code>\cleaders</code> | 2957 |
| <code>\@part</code> | 1878 | <code>\clearoublepage</code> | 79, 86 |
| <code>\@schapter</code> | 1902 | <code>\clearpage</code> | 62 |
| <code>\@secondoftwo</code> | 1434, 1513 | <code>\closein</code> | 2152 |
| <code>\@sect</code> | 1890 | <code>\columnsep</code> | 1307 |
| <code>\@spart</code> | 1896 | <code>\columnwidth</code> | 1306 |
| <code>\@ssect</code> | 1908 | <code>\comma@entry</code> | 344, 345, 347, 353, 379, 380, 382, 388, 615, 617, 621, 1632, 1633, 1634, 1640, 1643 |
| <code>\@stpelt</code> | 1725, 1726 | <code>\comma@parse</code> | 343, 378, 614, 1631 |
| <code>\@testdef</code> | 1334, 1335, 1495, 1496 | <code>\count@</code> | 1349, 1360, 1361, 1363, 1398, 1399, 1408, 1410, 1411, 1524, 1547, 1549, 1550, 1555, 1556, 2078, 2083, 2094, 2095 |
| <code>\@testopt</code> | 2110, 2113, 2123 | <code>\csname</code> | 253, 254, 291, 316, 317, 318, 327, 352, 353, 370, 371, 387, 388, 405, 406, 426, 428, 445, 463, 466, 479, 537, 539, 540, 545, 555, 560, 566, 579, 588, 605, 621, 652, 662, 731, 739, 745, 788, 789, 791, 811, 834, 835, 836, 897, 1339, 1500, 1661, 1669, 1721, 1750, 1752, 1755, 1757, 1769, 1775, 1780, 1781, 1783, 1785, 1786, 1793, 1846, 1849, 2087, 2094, 2106, 2130, 2234, 2236, 2251, 2252, 2269, 2276, 2279, 2293, 2294, 2314, 2411, 2434, 2436, 2607 |
| <code>\@tfor</code> | 323, 721 | <code>\current@chapid</code> | 80, 88 |
| <code>\@undefined</code> | 1793, 2362, 2465, 2489, 2497 | D | |
| <code>\@</code> | 25, 26, 27, 28, 153, 155, 157, 158, 170, 173, 2271, 2365, 2390, 2400, 2404, 2420 | <code>\DeclareBoolOption</code> | 2063, 2064, 2065, 2069 |
| <code>\@</code> | 44, 45 | <code>\DeclareOption</code> | 194 |
| A | | | |
| <code>\AddLineBeginAux</code> | 281 | | |
| <code>\advance</code> | 1061, 1399, 1556, 1762, 2084, 2933 | | |
| <code>\afterassignment</code> | 234, 1145, 1149 | | |
| <code>\AfterLastShipout</code> | 1058, 1416, 1588 | | |
| <code>\Alph</code> | 7 | | |
| <code>\anchor</code> | 2402 | | |
| <code>\AtBeginDocument</code> | 1036, 1234, 1690, 1870 | | |
| <code>\AtBeginShipout</code> | 1013, 1105 | | |
| <code>\AtBeginShipoutAddToBox</code> | 1106 | | |

| | | | |
|--|---|---|---|
| <code>\default</code> | 2383, 2423 | <code>\gdef</code> | 413, 540, 545, 946, 1339, 1697, 1750, 1752 |
| <code>\define@key</code> | 1845, 1848, 1851, 1854, 2066, 2917, 2920, 2923 | <code>\GetTitleStringDisableCommands</code> | 1841 |
| <code>\descriptionlabel</code> | 1914 | <code>\GetTitleStringExpand</code> | 1827 |
| <code>\detokenize</code> | 1832 | <code>\GetTitleStringNonExpand</code> | 1829 |
| <code>\dftest</code> | 167, 174, 175, 176, 177, 178, 179, 180, 181, 182 | <code>\GetTitleStringResult</code> | 1832 |
| <code>\dimen@</code> | 2938, 2944, 2957 | H | |
| <code>\dimexpr</code> 153, 155, 1411, 2791, 2820, 2944 | | <code>\hb@xt@</code> | 2957 |
| <code>\directlua</code> | 2465, 2489, 2497 | <code>\headheight</code> | 1301 |
| <code>\do</code> | 328, 721 | <code>\headmargin</code> | 1313 |
| <code>\documentclass</code> | 4, 39, 68, 273 | <code>\headsep</code> | 1302 |
| <code>\dotfill</code> | 169, 173 | <code>\hfill</code> | 2945, 2957 |
| E | | <code>\hoffset</code> | 1294 |
| <code>\emph</code> | 150 | <code>\hss</code> | 2957 |
| <code>\end</code> | 34, 64, 130, 159, 183, 185 | I | |
| <code>\endcsname</code> | 253, 254, 291, 316, 317, 318, 327, 352, 353, 370, 371, 387, 388, 405, 406, 426, 428, 445, 464, 466, 479, 537, 539, 540, 545, 555, 560, 566, 579, 588, 605, 621, 652, 662, 737, 739, 745, 788, 789, 791, 811, 834, 835, 836, 884, 897, 1339, 1500, 1661, 1669, 1720, 1721, 1727, 1750, 1752, 1755, 1756, 1757, 1769, 1775, 1780, 1781, 1783, 1785, 1786, 1793, 1846, 1849, 2087, 2094, 2106, 2130, 2234, 2236, 2251, 2252, 2269, 2276, 2279, 2293, 2294, 2314, 2411, 2434, 2437, 2607 | <code>\if@filesw</code> | 711, 1059, 1695, 2499, 2505, 2512 |
| <code>\endinput</code> | 192, 265, 278, 967, 1005, 1026, 1055, 1096, 1136, 1227, 1245, 1427, 1472, 1599, 1710, 1804, 2048, 2451, 2463, 2471, 2484, 2555, 2560, 2912, 2967 | <code>\if@safe@actives</code> | 890 |
| <code>\escapechar</code> | 330, 421, 461, 462, 468, 723, 1252 | <code>\ifcase</code> | 115, 1203, 1632, 2704, 2717, 2781, 2810 |
| <code>\etex@unexpanded</code> .. | 592, 820, 840, 2212 | <code>\ifcsname</code> | 884, 1720, 1727, 1756 |
| <code>\evensidemargin</code> | 1298 | <code>\ifeof</code> | 2137, 2191 |
| <code>\externaldocument</code> | 2238, 2280 | <code>\ifetex@unexpanded</code> | 268 |
| F | | <code>\IfFormatAtLeastTF</code> | 223, 1314 |
| <code>\fancyhead</code> | 51, 54 | <code>\ifheadnameref</code> | 1933, 1946 |
| <code>\fancyhf</code> | 50, 53 | <code>\ifin@</code> | 318 |
| <code>\fancypagestyle</code> | 52 | <code>\ifluatex</code> .. | 1271, 1431, 1509, 1522, 1565, 1581, 2689, 2715, 2770, 2800 |
| <code>\filename@area</code> | 2213 | <code>\ifnum</code> | 477, 1073, 1178, 1188, 1194, 1398, 1555, 1604, 1657, 2083, 2160, 2164, 2169, 2478, 2517, 2581, 2593, 2636, 2649, 2667, 2777, 2806, 2942 |
| <code>\filename@parse</code> | 2122 | <code>\ifodd</code> | 124, 2827, 2878 |
| <code>\fmtversion</code> | 223, 1028 | <code>\ifpdf</code> ... | 2474, 2766, 2771, 2796, 2801 |
| <code>\foo</code> | 18, 29, 31, 33 | <code>\ifx</code> 438, 442, 475, 508, 566, 674, 677, 691, 730, 796, 965, 970, 977, 1003, 1024, 1053, 1094, 1134, 1225, 1243, 1338, 1425, 1471, 1480, 1499, 1510, 1566, 1597, 1614, 1623, 1627, 1632, 1633, 1634, 1708, 1780, 1802, 1925, 1929, 1986, 2006, 2030, 2046, 2094, 2177, 2200, 2205, 2210, 2224, 2271, 2362, 2365, 2390, 2393, 2400, 2404, 2420, 2449, 2461, 2465, 2489, 2497, 2553, 2558, 2776, 2805, 2910, 2965, 2972 | |
| <code>\footskip</code> | 1303 | <code>\ifZREF@found</code> | 248, 2381, 2388 |
| <code>\foremargin</code> | 1310 | <code>\ifZREF@immediate</code> | 635, 701, 713, 717, 732 |
| <code>\frontmatter</code> | 58, 103 | <code>\ifZREF@pa@list</code> | 1487, 1492 |
| G | | <code>\ifZREF@pl@list</code> | 1326, 1331 |
| <code>\g@addto@macro</code> | 351, 369, 1755 | <code>\ifzref@titleref@expand</code> .. | 1810, 1826 |
| <code>\G@refundefinedtrue</code> | 777 | <code>\ifzref@titleref@stripperiod</code> ... | 1823, 1834 |

| | | | |
|---|---|--|---|
| <code>\number</code> | 94, 109, 916, 921, 955, 1112, 1257, 1258, 1273, 1274, 1277, 1278, 1281, 1282, 1285, 1286, 1691, 1744, 1770, 1777, 2566, 2568, 2573, 2575, 2580, 2592, 2650, 2662, 2827, 2878, 2935 | <code>\refstepcounter</code> | 1038 |
| <code>\numexpr</code> | 94, 109, 115, 923, 1114, 1175, 1691, 1744, 2219, 2230, 2264, 2349, 2358, 2585, 2597, 2791, 2820, 2942 | <code>\renewcommand</code> | 7, 46, 915 |
| O | | | |
| <code>\oddsidemargin</code> | 1297 | <code>\RequirePackage</code> | 191, 196, 224, 225, 226, 227, 228, 231, 267, 272, 280, 964, 1002, 1007, 1023, 1050, 1051, 1052, 1093, 1098, 1099, 1133, 1138, 1139, 1140, 1141, 1224, 1229, 1230, 1242, 1247, 1248, 1249, 1424, 1429, 1437, 1438, 1453, 1465, 1474, 1475, 1476, 1596, 1707, 1712, 1801, 1806, 1807, 2045, 2050, 2051, 2448, 2460, 2552, 2557, 2562, 2909, 2914, 2916, 2964 |
| <code>\on@line</code> | 1693, 2154 | <code>\reset@font</code> | 929 |
| <code>\openin</code> | 2136 | <code>\rightarrow</code> | 45 |
| P | | | |
| <code>\PackageError</code> | 259, 270, 295, 305, 490, 1605, 2467, 2479 | <code>\romannumeral</code> | 583, 805, 826, 1615, 2085, 2094, 2095 |
| <code>\PackageInfo</code> | 292, 521, 1694, 2143, 2155, 2246, 2288, 2309 | S | |
| <code>\PackageWarning</code> | 346, 364, 381, 399, 616, 680, 1635, 1676, 2138 | <code>\savepos</code> | 2506 |
| <code>\PackageWarningNoLine</code> | 2225, 2344, 2353 | <code>\section</code> | 63, 137, 145 |
| <code>\page</code> | 2386 | <code>\setcounter</code> | 1012 |
| <code>\pageheight</code> | 1281, 1282, 2717 | <code>\setlength</code> | 2938 |
| <code>\pagestyle</code> | 49 | <code>\SetupKeyvalOptions</code> | 2059 |
| <code>\pagewidth</code> | 1285, 1286 | <code>\ShipoutBoxDepth</code> | 1318 |
| <code>\paperheight</code> | 1266 | <code>\ShipoutBoxHeight</code> | 1317 |
| <code>\paperwidth</code> | 1265 | <code>\ShipoutBoxWidth</code> | 1316 |
| <code>\pdf@escapehex</code> | 1484 | <code>\space</code> | 779, 1448, 1460, 2157, 2158, 2162, 2166, 2171, 2468, 2480 |
| <code>\pdf@strcmp</code> | 477 | <code>\spinmargin</code> | 1309 |
| <code>\pdf@unescapehex</code> | 1485 | <code>\stepcounter</code> 19, 1014, 1718, 1719, 1729 | |
| <code>\pdfhorigin</code> | 1291 | <code>\stockheight</code> | 1268 |
| <code>\pdflastxpos</code> | 2490 | <code>\stockwidth</code> | 1267 |
| <code>\pdflastypos</code> | 2491 | T | |
| <code>\pdfpageattr</code> | 1448, 1454 | <code>\tableofcontents</code> | 59, 132 |
| <code>\pdfpageheight</code> | 1290, 2704 | <code>\textheight</code> | 1300 |
| <code>\pdfpagesattr</code> | 1460, 1466 | <code>\textwidth</code> | 1299 |
| <code>\pdfpagewidth</code> | 1289 | <code>\TeXeTstate</code> | 2517 |
| <code>\pdfsavepos</code> 2468, 2480, 2500, 2936, 2950 | | <code>\the</code> | 13, 153, 155, 430, 445, 461, 557, 562, 621, 627, 743, 757, 923, 1016, 1067, 1108, 1114, 1175, 1332, 1352, 1359, 1360, 1361, 1363, 1408, 1410, 1411, 1439, 1441, 1454, 1466, 1493, 1528, 1546, 1547, 1549, 1820, 2213, 2219, 2230, 2264, 2349, 2358, 2490, 2491, 2493, 2494 |
| <code>\pdfTeXversion</code> | 2478 | <code>\thechapter</code> | 14 |
| <code>\pdfvariable</code> | 1273, 1274, 1277, 1278, 1439, 1441 | <code>\thefoo</code> | 7, 12, 20 |
| <code>\pdfvorigin</code> | 1292 | <code>\thetype</code> | 2276 |
| <code>\ProcessOptions</code> | 217 | <code>\thepage</code> | 43, 44, 45, 714, 718, 779, 937, 1768 |
| <code>\protect</code> | 777 | <code>\thezpage</code> | 18, 1768, 1774 |
| <code>\protected</code> | 239 | <code>\thezref@unique</code> | 11, 915, 1765, 1766, 1773, 1774, 1776, 2937, 2939, 2942, 2943, 2951 |
| <code>\protected@write</code> | 716 | | |
| <code>\providecommand</code> | 223, 282, 1689, 2055, 2926, 2927, 2928 | | |
| <code>\ProvidesPackage</code> | 189, 221, 962, 1000, 1021, 1048, 1091, 1131, 1222, 1240, 1422, 1594, 1687, 1705, 1799, 2043, 2446, 2458, 2550, 2907, 2962 | | |
| R | | | |
| <code>\read</code> | 2189 | | |

| | | | |
|---------------------------------------|------------------------------------|---|---|
| <code>\title</code> | 2395, 2424 | <code>\zlistpagelayout</code> | 17, 1327 |
| <code>\toks@</code> | 424, | <code>\zmakeperpage</code> | 18, <u>1736</u> |
| | 430, 444, 445, 554, 557, 559, | <code>\znextpage</code> | 15, 51, 54, 1158 |
| | 562, 613, 620, 621, 627, 741, | <code>\znextpagesetup</code> | 15, 42, 1144 |
| | 743, 756, 757, 1062, 1067, 1348, | <code>\znonextpagename</code> | 46, 1161, 1209 |
| | 1352, 1358, 1359, 1521, 1528, | <code>\zpageref</code> | 12, 126, <u>991</u> |
| | 1545, 1546, 1814, 1820, 2198, 2213 | <code>\zposx</code> | 19, 153, <u>2540</u> , 2942, 2943 |
| <code>\topmargin</code> | 1296 | <code>\zposy</code> | 19, 155, <u>2540</u> |
| <code>\TR@TitleReference</code> .. | 2320, 2380, 2419 | <code>\zref</code> | 12, 25, 26, 27, 28, 112, |
| <code>\trimedge</code> | 1308 | | 114, 123, 128, 129, 139, <u>982</u> , 992 |
| <code>\trimtop</code> | 1311 | <code>\ZREF@@@delprop</code> ... | 435, <u>437</u> , <u>474</u> , <u>474</u> |
| <code>\ttl@sect@i</code> | 1978 | <code>\ZREF@@@newprop</code> | 539, 543 |
| <code>\ttl@straight@i</code> | 1984 | <code>\ZREF@@delprop</code> | |
| | | | 425, <u>434</u> , 448, 465, 471, 481 |
| | | <code>\ZREF@@extract</code> | 809, <u>815</u> |
| | | <code>\ZREF@@makeperpage</code> .. | 1738, 1744, 1748 |
| | | <code>\ZREF@@newprop</code> | 517, 528, 531, <u>535</u> |
| | | <code>\ZREF@@perpage@step</code> | 1753, <u>1761</u> |
| | | <code>\ZREF@abspos@body@x@center</code> | <u>2857</u> |
| | | <code>\ZREF@abspos@body@x@left</code> . | <u>2855</u> , <u>2882</u> |
| | | <code>\ZREF@abspos@body@x@right</code> .. | <u>2856</u> , <u>2879</u> |
| | | <code>\ZREF@abspos@body@y@bottom</code> | |
| | | | <u>2862</u> , <u>2874</u> , <u>2896</u> |
| | | <code>\ZREF@abspos@body@y@center</code> .. | <u>2866</u> , <u>2897</u> |
| | | <code>\ZREF@abspos@body@y@top</code> | |
| | | | <u>2858</u> , <u>2863</u> , <u>2867</u> , <u>2895</u> |
| | | <code>\ZREF@abspos@foot@x@center</code> | <u>2872</u> |
| | | <code>\ZREF@abspos@foot@x@left</code> | <u>2870</u> |
| | | <code>\ZREF@abspos@foot@x@right</code> | <u>2871</u> |
| | | <code>\ZREF@abspos@foot@y@bottom</code> | <u>2873</u> |
| | | <code>\ZREF@abspos@head@x@center</code> | |
| | | | <u>2837</u> , <u>2857</u> , <u>2872</u> |
| | | <code>\ZREF@abspos@head@x@left</code> | |
| | | | <u>2823</u> , <u>2834</u> , <u>2838</u> , <u>2855</u> , <u>2870</u> |
| | | <code>\ZREF@abspos@head@x@right</code> | |
| | | | <u>2833</u> , <u>2856</u> , <u>2871</u> |
| | | <code>\ZREF@abspos@head@y@bottom</code> .. | <u>2847</u> , <u>2859</u> |
| | | <code>\ZREF@abspos@head@y@center</code> | <u>2851</u> |
| | | <code>\ZREF@abspos@head@y@top</code> | |
| | | | <u>2841</u> , <u>2848</u> , <u>2852</u> |
| | | <code>\ZREF@abspos@marginpar@x@center</code> .. | <u>2891</u> |
| | | <code>\ZREF@abspos@marginpar@x@left</code> .. | |
| | | | <u>2877</u> , <u>2888</u> , <u>2892</u> |
| | | <code>\ZREF@abspos@marginpar@x@right</code> .. | <u>2887</u> |
| | | <code>\ZREF@abspos@marginpar@y@bottom</code> .. | <u>2896</u> |
| | | <code>\ZREF@abspos@marginpar@y@center</code> .. | <u>2897</u> |
| | | <code>\ZREF@abspos@marginpar@y@top</code> .. | <u>2895</u> |
| | | <code>\ZREF@abspos@media@height</code> | |
| | | | <u>2692</u> , <u>2738</u> , <u>2744</u> |
| | | <code>\ZREF@abspos@media@width</code> | |
| | | | <u>2674</u> , <u>2731</u> , <u>2735</u> |
| | | <code>\ZREF@abspos@media@x@center</code> ... | <u>2733</u> |
| | | <code>\ZREF@abspos@media@x@left</code> .. | <u>2727</u> , <u>2734</u> |
| | | <code>\ZREF@abspos@media@x@right</code> | <u>2730</u> |
| | | <code>\ZREF@abspos@media@y@bottom</code> ... | <u>2740</u> |
| | | <code>\ZREF@abspos@media@y@center</code> ... | <u>2743</u> |
| | | <code>\ZREF@abspos@media@y@top</code> . | <u>2737</u> , <u>2755</u> |
| | | <code>\ZREF@abspos@origin@x</code> ... | <u>2764</u> , <u>2825</u> |
| U | | | |
| <code>\unexpanded</code> | 271, 276 | | |
| <code>\UniqueCounterCall</code> | 1159 | | |
| <code>\UniqueCounterNew</code> | 1142 | | |
| <code>\uppermargin</code> | 1312 | | |
| <code>\url</code> | 2415 | | |
| <code>\urluse</code> | 2410 | | |
| <code>\usepackage</code> | 9, 41, 48, 70, 72 | | |
| V | | | |
| <code>\value</code> | 13, 1108, 1332, 1493 | | |
| <code>\verb</code> | 173 | | |
| <code>\voffset</code> | 1295 | | |
| W | | | |
| <code>\write</code> | 705, 706, 1696 | | |
| X | | | |
| <code>\x</code> | 331, 336, 724, 729, 898, 900, | | |
| | 1201, 1217, 1254, 1261, 1351, | | |
| | 1354, 1357, 1397, 1526, 1531, | | |
| | 1544, 1554, 1659, 1664, 1667, | | |
| | 1673, 1743, 1746, 2024, 2025, | | |
| | 2030, 2197, 2200, 2205, 2210, | | |
| | 2220, 2223, 2233, 2234, 2265, 2270 | | |
| <code>\XR@ext</code> | 2056 | | |
| Y | | | |
| <code>\y</code> | 2026, 2027 | | |
| Z | | | |
| <code>\z</code> | 2027, 2028, 2030 | | |
| <code>\z@</code> | 2953 | | |
| <code>\Z@D@page</code> | 1162 | | |
| <code>\Z@L@LastPage</code> | 1064 | | |
| <code>\Z@L@main</code> | 1063 | | |
| <code>\Z@L@ZREF@temp</code> | 633, 637, 640, 651, 661 | | |
| <code>\zdotfill</code> | 20, 170, 173, <u>2930</u> | | |
| <code>\zdotfillsetup</code> | 20, <u>2929</u> | | |
| <code>\zexternaldocument</code> | 21, <u>2101</u> | | |
| <code>\ziflastpage</code> | 13, <u>1080</u> | | |
| <code>\zifrefundefined</code> | 8, <u>763</u> | | |
| <code>\zkvlabel</code> | <u>976</u> | | |
| <code>\zlabel</code> | 12, 83, 104, 138, 146, <u>969</u> | | |
| <code>\zlistpageattr</code> | 1488 | | |

| | | | |
|--|--|---|--|
| <code>\ZREF@abspos@origin@y</code> . . . | 2794 , 2843 | <code>\ZREF@dotfill</code> | 2940 , 2947 , 2956 |
| <code>\ZREF@abspos@paper@x@center</code> | 2752 , 2900 | <code>\ZREF@ENV@line</code> | 2975 , 2978 |
| <code>\ZREF@abspos@paper@x@left</code> | 2746 , 2824 , 2898 | <code>\zref@env@line</code> | 2970 , 2971 |
| <code>\ZREF@abspos@paper@x@right</code> 2749 , 2899 | | <code>\ZREF@extract</code> | 804 , 821 , 824 , 876 |
| <code>\ZREF@abspos@paper@y@bottom</code> | 2756 , 2902 | <code>\zref@extract</code> | 8 , 95 , 96 , 109 , 140 , 804 , 824 , 853 , 871 , 876 , 989 , 1117 , 1213 , 1363 , 1410 , 1411 , 1535 , 1572 , 1773 , 1774 , 1867 , 2541 , 2544 , 2731 , 2735 , 2738 , 2744 , 2750 , 2753 , 2758 , 2762 , 2783 , 2812 , 2826 , 2828 , 2830 , 2835 , 2839 , 2844 , 2845 , 2849 , 2853 , 2860 , 2864 , 2868 , 2875 , 2880 , 2883 , 2884 , 2889 , 2893 |
| <code>\ZREF@abspos@paper@y@center</code> | 2760 , 2903 | <code>\ZREF@extractdefault</code> 825 , 841 , 844 , 875 | |
| <code>\ZREF@abspos@paper@y@top</code> | 2755 , 2757 , 2761 , 2842 , 2901 | <code>\zref@extractdefault</code> . | 8 , 116 , 117 , 817 , 844 , 864 , 870 , 875 , 1073 , 1074 , 1171 , 1186 , 1232 , 1776 , 2318 , 2321 , 2322 , 2326 , 2327 , 2330 , 2332 , 2333 , 2335 , 2337 , 2568 , 2575 , 2634 , 2662 , 2827 , 2878 |
| <code>\ZREF@abspos@stock@x@center</code> . . . | 2900 | <code>\ZREF@false</code> | 677 , 687 |
| <code>\ZREF@abspos@stock@x@left</code> | 2898 | <code>\ZREF@foundfalse</code> | 2379 |
| <code>\ZREF@abspos@stock@x@right</code> | 2899 | <code>\ZREF@foundtrue</code> | 2426 |
| <code>\ZREF@abspos@stock@y@bottom</code> . . . | 2902 | <code>\ZREF@getcurrent</code> . . | 582 , 593 , 596 , 874 |
| <code>\ZREF@abspos@stock@y@center</code> . . . | 2903 | <code>\zref@getcurrent</code> | 7 , 596 , 869 , 874 |
| <code>\ZREF@abspos@stock@y@top</code> | 2901 | <code>\zref@hex</code> | 1439 , 1441 , 1454 , 1466 , 1481 , 1484 |
| <code>\ZREF@abspos@used</code> | 2626 , 2628 | <code>\zref@ifabsposnumundefined</code> | 2661 , 2666 |
| <code>\ZREF@absposnum</code> | 2585 , 2597 , 2603 | <code>\zref@ifabsposundefined</code> | 2659 |
| <code>\zref@absposnumused</code> | 2648 | <code>\ZREF@ifDefinable</code> | 242 , 763 , 991 , 994 , 1080 , 1122 , 1158 , 1327 , 1488 , 1736 , 1790 , 1857 , 1860 , 2101 , 2525 , 2530 , 2535 , 2930 |
| <code>\zref@absposnumx</code> . . . | 2567 , 2579 , 2617 | <code>\ZREF@iflastpage</code> . . . | 1081 , 1083 , 1083 |
| <code>\zref@absposnumy</code> . . . | 2574 , 2591 , 2620 | <code>\zref@iflastpage</code> | 13 , 1072 , 1086 |
| <code>\zref@absposused</code> | 2625 | <code>\zref@iflistcontainsprop</code> | 6 , 310 , 345 , 363 , 380 , 398 , 648 , 658 |
| <code>\zref@absposx</code> | 2565 , 2611 | <code>\zref@iflistundefined</code> | 6 , 289 , 300 , 304 , 311 |
| <code>\zref@absposy</code> | 2572 , 2614 | <code>\zref@ifpropundefined</code> | 7 , 485 , 489 , 519 , 548 , 615 , 828 , 1406 , 1656 , 2373 |
| <code>\zref@addprop</code> | 6 , 76 , 360 , 1017 , 1031 , 1035 , 1101 , 1104 , 1259 , 1275 , 1279 , 1283 , 1287 , 1440 , 1442 , 1455 , 1467 , 1643 , 1809 , 2454 , 2563 , 2564 | <code>\ZREF@ifrefcontainsprop</code> . . . | 787 , 795 |
| <code>\zref@addprops</code> | 6 , 15 , 341 , 938 , 1325 , 1715 , 2496 | <code>\zref@ifrefcontainsprop</code> | 9 , 783 , 1408 , 2334 , 2430 , 2431 |
| <code>\ZREF@addtoks</code> | 755 | <code>\ZREF@ifrefundefined</code> | 765 , 768 , 1168 , 1179 , 1189 |
| <code>\ZREF@base@ok</code> | 958 , 965 , 1003 , 1024 , 1053 , 1094 , 1134 , 1225 , 1243 , 1425 , 1597 , 1708 , 1802 , 2046 , 2449 , 2461 , 2553 , 2910 , 2965 | <code>\zref@ifrefundefined</code> | 8 , 760 , 770 , 776 , 784 , 827 , 1180 , 1361 , 1549 , 1766 , 2582 , 2594 , 2630 , 2660 , 2668 , 2939 |
| <code>\ZREF@call</code> 1167 , 1182 , 1191 , 1195 , 1203 | | <code>\ZREF@immediatefalse</code> | 678 |
| <code>\ZREF@def@abspos</code> | 2611 , 2614 , 2617 , 2620 , 2622 | <code>\ZREF@immediatetrue</code> | 675 , 704 |
| <code>\zref@def@absposnumx</code> | 2616 | <code>\ZREF@label</code> 603 , 627 , 637 , 640 , 710 , 1067 | |
| <code>\ZREF@def@absposnumy</code> | 2622 | <code>\zref@label</code> | 7 , 597 , 973 |
| <code>\zref@def@absposnumy</code> | 2619 | <code>\zref@labelbykv</code> | 630 , 980 |
| <code>\zref@def@absposx</code> | 2610 | <code>\zref@labelbylist</code> | 7 , 598 , 600 , 1108 , 1765 , 2527 |
| <code>\zref@def@absposy</code> | 2613 | | |
| <code>\ZREF@def@extract</code> | 846 , 848 | | |
| <code>\zref@def@extract</code> | 8 , 845 | | |
| <code>\ZREF@def@extractdefault</code> . . . | 857 , 859 | | |
| <code>\zref@def@extractdefault</code> | 856 | | |
| <code>\ZREF@default</code> | 562 , 563 , 572 | | |
| <code>\zref@default</code> . . . | 9 , 531 , 807 , 926 , 928 | | |
| <code>\ZREF@delprop</code> | 413 , 416 , 418 , 453 , 456 , 458 | | |
| <code>\zref@delprop</code> | 412 , 452 | | |
| <code>\ZREF@df@dot</code> | 2924 , 2928 , 2957 | | |
| <code>\ZREF@df@min</code> | 2921 , 2926 , 2944 | | |
| <code>\ZREF@df@unit</code> | 2918 , 2927 , 2938 | | |

| | | | |
|---|---|---|---|
| <code>\zref@labelbyprops</code> | 7, 88, 610, 1166, 2532, 2537, 2937, 2951 | <code>\ZREF@org@@schapter</code> | 1904 |
| <code>\zref@listexists</code> | 6, 303, 322, 342, 361, 377, 396, 419, 459, 602 | <code>\ZREF@org@@sect</code> | 1892 |
| <code>\zref@listforloop</code> | 321, 657 | <code>\ZREF@org@@spart</code> | 1898 |
| <code>\zref@listpageattr</code> | 1488 | <code>\ZREF@org@@ssect</code> | 1910 |
| <code>\zref@listpagelayout</code> | 1327 | <code>\ZREF@org@@stpelt</code> | 1725, 1730, 1734 |
| <code>\zref@localaddprop</code> | 395 | <code>\ZREF@org@beamer@section</code> | 1960 |
| <code>\zref@localaddprops</code> | 376 | <code>\ZREF@org@beamer@subsection</code> | 1966 |
| <code>\zref@localdelprop</code> | 415, 455, 669 | <code>\ZREF@org@beamer@subsubsection</code> | 1972 |
| <code>\ZREF@mainlist</code> | 598, 932, 935, 938, 1017, 1031, 1035, 1809, 2454 | <code>\ZREF@org@descriptionlabel</code> | 1916 |
| <code>\ZREF@makeperpage@opt</code> | 1738, 1741 | <code>\ZREF@org@lst@MakeCaption</code> | 2012 |
| <code>\ZREF@MARKS@DefineProp</code> | 1619, 1620, 1621, 1655 | <code>\ZREF@org@LT@cc@ption</code> | 1998 |
| <code>\zref@marks@register</code> | 1601, 1606, 1638, 1677 | <code>\ZREF@org@M@sect</code> | 1951 |
| <code>\ZREF@name</code> | 229, 259, 270, 292, 295, 305, 346, 364, 381, 399, 490, 509, 521, 525, 549, 564, 616, 680, 692, 1605, 2467, 2479 | <code>\ZREF@org@refstepcounter</code> | 1040 |
| <code>\ZREF@NAME@bot</code> | 1634, 1654 | <code>\ZREF@org@stepcounter</code> | 1718, 1723, 1729 |
| <code>\ZREF@NAME@first</code> | 1633, 1653 | <code>\ZREF@org@testdef</code> | 1334, 1336, 1495, 1497 |
| <code>\ZREF@NAME@top</code> | 1632, 1652 | <code>\ZREF@org@thepage</code> | 714, 718 |
| <code>\zref@NewLabel</code> | 8, 282, 285, 750, 2217, 2303 | <code>\ZREF@org@ttl@sect@i</code> | 1980 |
| <code>\zref@newlist</code> | 6, 288, 935, 1057, 1100, 1629, 1714, 2488 | <code>\ZREF@org@ttl@straight@i</code> | 1991 |
| <code>\ZREF@newprop</code> | 498, 501, 504 | <code>\ZREF@org@write</code> | 705, 706 |
| <code>\zref@newprop</code> | 7, 12, 13, 14, 75, 495, 936, 937, 940, 947, 951, 955, 1016, 1030, 1034, 1256, 1272, 1276, 1280, 1284, 1316, 1317, 1318, 1321, 1322, 1323, 1439, 1441, 1454, 1466, 1660, 1668, 2052, 2053, 2054, 2374, 2490, 2491, 2493, 2494, 2969, 2970 | <code>\ZREF@P</code> | 505, 506, 508, 510, 519, 522, 526, 536, 537, 539, 540, 541, 545, 721, 725, 726, 735, 739, 744, 745 |
| <code>\ZREF@NewPropAnchor</code> | 939, 2103, 2453 | <code>\ZREF@pa@AfterLastShipout</code> | 1491, 1589 |
| <code>\ZREF@NewPropPageValue</code> | 954, 1102, 1713 | <code>\ZREF@pa@AtVeryEnd</code> | 1504, 1507, 1578 |
| <code>\ZREF@NewPropTheotype</code> | 950, 2274 | <code>\ZREF@pa@ListPage</code> | 1525, 1543 |
| <code>\ZREF@NewPropTitle</code> | 945, 1808, 2104 | <code>\ZREF@pa@listtrue</code> | 1489 |
| <code>\ZREF@nextpage</code> | 1159, 1163 | <code>\ZREF@page@max</code> | 1332, 1398, 1493, 1555 |
| <code>\ZREF@nil</code> | 545, 797, 836, 2190, 2196, 2202, 2207, 2217, 2233, 2262, 2270, 2361, 2368, 2377, 2380, 2419 | <code>\zref@pageattr</code> | 1533 |
| <code>\ZREF@NOVALUE</code> | 803 | <code>\zref@pageattr@used</code> | 1540 |
| <code>\ZREF@novalue</code> | 796, 797, 803 | <code>\ZREF@pagenum@last</code> | 1185, 1188 |
| <code>\ZREF@np@call@next</code> | 1153, 1157, 1212 | <code>\ZREF@pagenum@this</code> | 1170, 1175, 1178, 1188, 1194 |
| <code>\ZREF@np@call@nonext</code> | 1150, 1156, 1208 | <code>\ZREF@par</code> | 508, 533 |
| <code>\ZREF@np@call@unknown</code> | 1146, 1155, 1204 | <code>\ZREF@param</code> | 422, 423, 442, 460, 477, 646, 647, 648, 652, 673, 674, 677, 682 |
| <code>\ZREF@np@setup@i</code> | 1145, 1148 | <code>\ZREF@patch</code> | 249, 1037, 1871, 1877, 1883, 1889, 1895, 1901, 1907, 1913, 1944, 1957, 1963, 1969, 1977, 1983, 1996, 2004, 2016, 2031 |
| <code>\ZREF@np@setup@ii</code> | 1149, 1152 | <code>\zref@pdfpageattr</code> | 1450, 1533, 1539, 1550 |
| <code>\ZREF@number</code> | 920, 1535, 1541, 1603, 2667 | <code>\zref@pdfpageattr@used</code> | 1451 |
| <code>\ZREF@org@@begintheorem</code> | 2034 | <code>\zref@pdfpagesattr</code> | 1462, 1570, 1583 |
| <code>\ZREF@org@@caption</code> | 1874 | <code>\zref@pdfpagesattr@used</code> | 1463, 1575 |
| <code>\ZREF@org@@chapter</code> | 1886, 1942 | <code>\ZREF@pl@AfterLastShipout</code> | 1330, 1417 |
| <code>\ZREF@org@@opargbegintheorem</code> | 2019 | <code>\ZREF@pl@AtVeryEnd</code> | 1343, 1346 |
| <code>\ZREF@org@@@part</code> | 1880 | <code>\ZREF@pl@ListEntry</code> | 1365, 1366, 1367, 1368, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1390, 1391, 1392, 1393, 1394, 1405 |

| | | | |
|------------------------------------|---|--|--|
| <code>\zref@pos@label@used</code> | 2640 | <code>\ZREF@TempNum</code> | 1603, 1604, 1608, 1615, 1657, 1670 |
| <code>\zref@pos@num@used</code> | 2654 | <code>\zref@thepage</code> | 14, 1116, 1125 |
| <code>\zref@prop</code> | 324, 332, 333, 337, 658, 662 | <code>\zref@thepage@atbegshi@hook</code> | 1103, 1107 |
| <code>\zref@propexists</code> | 7, 344, 362, 379, 397, 488, 578, 647, 668, 983 | <code>\zref@thepage@name</code> | 14, 1111, 1117, 1120, 1174 |
| <code>\ZREF@refname@next</code> | 1173, 1180, 1189, 1213 | <code>\zref@thepage@refused</code> | 1119, 1124 |
| <code>\ZREF@refname@this</code> | 1165, 1166, 1168, 1171 | <code>\ZREF@titleref</code> | 1861, 1863 |
| <code>\ZREF@RefPrefix</code> | 284, 286, 1338, 1499 | <code>\zref@titleref@cleanup</code> | 1812, 1852 |
| <code>\ZREF@refused</code> | 773, 775 | <code>\zref@titleref@current</code> | 945, 1831, 1835, 1836, 1855 |
| <code>\zref@refused</code> | 8, 769, 772, 849, 860, 988, 995, 1084, 1085, 1120, 1235, 1541, 1576, 1865, 2629, 2637, 2650 | <code>\ZREF@titleref@hook</code> | 1811, 1815, 1819, 1842 |
| <code>\zref@require@unique</code> | 11, 909, 1717, 2915 | <code>\zref@titleref@setcurrent</code> | 1825, 1873, 1879, 1885, 1891, 1897, 1903, 1909, 1915, 1923, 1926, 1930, 1934, 1936, 1947, 1949, 1959, 1965, 1971, 1979, 1987, 1989, 1999, 2008, 2018, 2033 |
| <code>\ZREF@Robust</code> | 232, 238, 244, 285, 288, 303, 310, 341, 360, 376, 395, 412, 415, 452, 455, 488, 495, 547, 577, 597, 600, 610, 630, 702, 772, 845, 856, 867, 883, 909, 925, 931, 1119, 1540, 1575, 1812, 1825, 2610, 2613, 2616, 2619, 2625 | <code>\zref@titleref@stripperperiodtrue</code> | 1824 |
| <code>\ZREF@SavedEscapechar</code> | 461, 468 | <code>\ZREF@true</code> | 674, 688 |
| <code>\zref@savepos</code> | 20, 2497, 2513, 2519 | <code>\ZREF@u@getcurrent</code> | 591 |
| <code>\ZREF@savepos@ok</code> | 2546, 2558 | <code>\zref@unhex</code> | 1482, 1485, 1534, 1571 |
| <code>\zref@setcurrent</code> | 7, 81, 541, 577, 697, 1039 | <code>\ZREF@UpdatePdfTeX</code> | 247, 2470, 2483 |
| <code>\zref@setdefault</code> | 9, 925, 928 | <code>\ZREF@value</code> | 557, 558, 571 |
| <code>\zref@setmainlist</code> | 9, 931 | <code>\ZREF@wrapper@babel</code> | 900, 906 |
| <code>\zref@showprop</code> | 547 | <code>\zref@wrapper@babel</code> | 11, 140, 765, 773, 846, 857, 883, 973, 980, 984, 1081, 1861, 2611, 2614, 2617, 2620, 2626 |
| <code>\ZREF@STAR</code> | 1627, 1651 | <code>\zref@wrapper@immediate</code> | 11, 87, 636, 702, 1066 |
| <code>\ZREF@stripperperiod</code> | 1836, 1844 | <code>\ZREF@wrapper@unexpanded</code> | 867, 881 |
| <code>\ZREF@temp</code> | 193, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 233, 234, 440, 441, 442, 720, 741, 742, 750, 1250, 1264, 1265, 1266, 1267, 1268, 1269, 1270, 1289, 1290, 1291, 1292, 1294, 1295, 1296, 1297, 1298, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, 1311, 1312, 1313, 1337, 1338, 1430, 1443, 1456, 1468, 1471, 1477, 1478, 1479, 1480, 1498, 1499, 1507, 1508, 1509, 1510, 1563, 1564, 1565, 1566, 1626, 1627, 1985, 1986, 2077, 2087, 2090, 2094, 2633, 2636, 2637, 2764, 2767, 2772, 2776, 2778, 2781, 2794, 2797, 2802, 2805, 2807, 2810, 2935 | <code>\zref@wrapper@unexpanded</code> | 11, 868, 873, 878, 2313 |
| <code>\ZREF@TempName</code> | 1602, 1614, 1615, 1617, 1643, 1656, 1660, 1668, 1679 | <code>\ZREF@wu@extract</code> | 819, 871 |
| | | <code>\ZREF@wu@extractdefault</code> | 839, 870 |
| | | <code>\ZREF@wu@getcurrent</code> | 591, 869 |
| | | <code>\ZREF@X</code> | 497, 500, 537 |
| | | <code>\zref@xr@</code> | 2067 |
| | | <code>\ZREF@xr@@AddUrl</code> | 2079, 2082 |
| | | <code>\ZREF@xr@@input</code> | 2210, 2305 |
| | | <code>\ZREF@xr@AddURL</code> | 2075, 2128, 2407 |
| | | <code>\ZREF@xr@checkfile</code> | 2132, 2135, 2185 |
| | | <code>\ZREF@xr@checkkey</code> | 2363, 2372 |
| | | <code>\ZREF@xr@checklist</code> | 2233, 2361 |
| | | <code>\zref@xr@ext</code> | 22, 2055, 2123 |
| | | <code>\ZREF@xr@externaldocument</code> | 2110, 2113, 2117 |
| | | <code>\ZREF@xr@externalfile</code> | 2120, 2121, 2239, 2281 |
| | | <code>\ZREF@xr@file</code> | 2121, 2136, 2139, 2145, 2156, 2179, 2227, 2346, 2355 |
| | | <code>\ZREF@xr@filelist</code> | 2119, 2177, 2180, 2182, 2183, 2211, 2212 |
| | | <code>\ZREF@xr@found</code> | 2147, 2157, 2219, 2264 |
| | | <code>\ZREF@xr@graburl</code> | 2123, 2125 |

| | |
|-----------------------------|--|
| \ZREF@xr@ignored@empty | 2252, 2267, 2269, 2276, 2279, 2294 |
| ... | 2148, 2160, 2162, 2229, 2230 |
| \ZREF@xr@ignored@ltx | 2085, 2087, 2089, 2095, 2130, 2411 |
| ... | 2150, 2169, 2171, 2357, 2358 |
| \ZREF@xr@ignored@zref | 2258, 2300, 2307 |
| ... | 2149, 2164, 2166, 2348, 2349 |
| \ZREF@xr@line | 2073, 2083, 2084, 2085 |
| ... | 2189, 2190, 2202, 2207 |
| \ZREF@xr@list | 2126, 2128, 2129, 2437 |
| ... | 2223, 2224 |
| \ZREF@xr@ltx@ignorewarning | 2242, 2284, 2429 |
| ... | 2352 |
| \ZREF@xr@newlabel | 2254, 2296, 2343 |
| ... | 2205, 2304 |
| \ZREF@xr@prefix | 2200, 2303 |
| ... | 2118, 2218, 2254, 2258, 2263, 2289, 2296, 2300 |
| \ZREF@xr@process@label | 2109 |
| ... | 2207, 2262 |
| \ZREF@xr@process@zreflabel | 2112 |
| ... | 2202, 2217 |
| \ZREF@xr@processfile | 984, 987 |
| ... | 2135, 2188 |
| \ZREF@xr@processline | 2510, 2527, 2532, 2537 |
| ... | 2190, 2196 |
| \ZREF@xr@refname | 12, 92, 93, 161, 162, 163, 994 |
| ... | 2218, 2244, 2251, 2263, 2286, 2293 |
| \ZREF@xr@relax | 17, 1689 |
| ... | 2306, 2393 |
| \ZREF@xr@scanparams | 19, 157, 158, 2525 |
| ... | 2268, 2377 |
| \ZREF@xr@scantitleref | 19, 2530 |
| ... | 2380, 2419 |
| \ZREF@xr@temp | 14, 1122 |
| ... | 2392, 2393 |
| \ZREF@xr@tempname | 18, 1860 |
| ... | 2221, 2222, 2242, 2247, 2258, 2266, 2267, 2284, 2300 |
| \ZREF@xr@temprefname | 19, 1845 |
| ... | 2222, 2234, 2236 |
| \ZREF@xr@theURL | 16, 124, 1231 |
| ... | 2085, 2087, 2089, 2095, 2130, 2411 |
| \ZREF@xr@tolabel | 15, 1162, 1205 |
| ... | 2258, 2300, 2307 |
| \ZREF@xr@URL | 18, 1790 |
| ... | 2073, 2083, 2084, 2085 |
| \ZREF@xr@url | 21, 2070 |
| ... | 2126, 2128, 2129, 2437 |
| \ZREF@xr@urlcheck | |
| ... | 2242, 2284, 2429 |
| \ZREF@xr@zref@ignorewarning | |
| ... | 2254, 2296, 2343 |
| \ZREF@xr@zref@newlabel | |
| ... | 2200, 2303 |
| \ZREF@xr@zreflabelfalse | |
| ... | 2109 |
| \ZREF@xr@zreflabeltrue | |
| ... | 2112 |
| \ZREF@zref | |
| ... | 984, 987 |
| \ZREF@zsavepos | |
| ... | 2510, 2527, 2532, 2537 |
| \zrefused | |
| ... | 12, 92, 93, 161, 162, 163, 994 |
| \zruns | |
| ... | 17, 1689 |
| \zsavepos | |
| ... | 19, 157, 158, 2525 |
| \zsaveposx | |
| ... | 19, 2530 |
| \zsaveposy | |
| ... | 2535 |
| \zthepage | |
| ... | 14, 1122 |
| \ztitleref | |
| ... | 18, 1860 |
| \ztitlerefsetup | |
| ... | 19, 1845 |
| \ztotpages | |
| ... | 16, 124, 1231 |
| \zunknownnextpagename | |
| ... | 15, 1162, 1205 |
| \zunmakeperpage | |
| ... | 18, 1790 |
| \zxrsetup | |
| ... | 21, 2070 |