The \texttt{zref} package

Heiko Oberdiek\textsuperscript{*}

\texttt<[heiko.oberdiek at googlemail.com>}

2016/05/21 v2.26

Abstract

Package \texttt{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 \hspace{1em} \textbf{Introduction} \hspace{1em} 4
  1.1 Standard \LaTeX{} behaviour \hspace{1em} 4
  1.2 Basic idea \hspace{1em} 4
  1.3 Interfaces \hspace{1em} 5

2 \hspace{1em} \textbf{Interface for programmers} \hspace{1em} 5
  2.1 Entities \hspace{1em} 5
  2.2 Property list \hspace{1em} 6
  2.3 Property \hspace{1em} 6
  2.4 Reference generation \hspace{1em} 7
  2.5 Data extraction \hspace{1em} 7
  2.6 Setup \hspace{1em} 8
  2.7 Declared properties \hspace{1em} 9
  2.8 Wrapper for advanced situations \hspace{1em} 10
  2.9 Counter for unique names \hspace{1em} 10

3 \hspace{1em} \textbf{User interface} \hspace{1em} 10
  3.1 Module \texttt{user} \hspace{1em} 10
  3.2 Module \texttt{abspage} \hspace{1em} 11
  3.3 Module \texttt{lastpage} \hspace{1em} 11
    3.3.1 Tests for last page \hspace{1em} 12
    3.3.2 Example \hspace{1em} 12
  3.4 Module \texttt{thepage} \hspace{1em} 13
  3.5 Module \texttt{nextpage} \hspace{1em} 13
    3.5.1 Configuration \hspace{1em} 13
    3.5.2 Example \hspace{1em} 14
  3.6 Module \texttt{totpages} \hspace{1em} 14
  3.7 Module \texttt{pagelayout} \hspace{1em} 15
  3.8 Module \texttt{marks} \hspace{1em} 15
  3.9 Module \texttt{runs} \hspace{1em} 16
  3.10 Module \texttt{perpage} \hspace{1em} 16
  3.11 Module \texttt{counter} \hspace{1em} 16
  3.12 Module \texttt{titleref} \hspace{1em} 17

\textsuperscript{*}Please report any issues at \url{https://github.com/ho-tex/oberdiek/issues}
3.13 Module \texttt{savepos} .................................................. 17
3.14 Module \texttt{dotfill} .................................................. 18
3.15 Module \texttt{env} .................................................. 18
3.16 Module \texttt{xr} .................................................. 18

4 Todo .......................................................... 19

5 Example .................................................. 19

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

Standard \LaTeX's reference system with \texttt{\label}, \texttt{\ref}, and \texttt{\pageref} supports two properties, the appearance of the counter that is last incremented by \texttt{\refstepcounter} and the page with the \texttt{\label} command.

Unhappily \LaTeX\ does not provide an interface for adding another property. Packages such as \texttt{hyperref}, \texttt{nameref}, or \texttt{titleref} are forced to use ugly hacks to extend the reference system. These ugly hacks are one of the causes for \texttt{hyperref}'s difficulty regarding compatibility with other packages.

1.1 Standard \LaTeX\ behaviour

References are created by the \texttt{\label} command:

\begin{verbatim}
\chapter{Second chapter}
\section{First section on page 7} \% section 2.1
\label{myref}
\end{verbatim}

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

\begin{verbatim}
\r@myref \rightarrow \{2.1\}\{7\}
\end{verbatim}

The length of the list is fixed in the \LaTeX\ kernel. An interface for adding new properties is missing.

There are several tries to add new properties:

\texttt{hyperref} uses a list of five properties instead of the standard list with two entries. This causes many compatibility problems with \LaTeX\ and other packages.

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

1.2 Basic idea

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

\begin{verbatim}
\g_xref_mylabel_plist \rightarrow
  \xref_dance_key{salsa}\xref_name_key{Morten}...
\end{verbatim}

The entries have the following format:
This approach is much more flexible:

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

Unhappily I am not familiar with the experimental code for \LaTeX3 that will need some time before its first release. Thus I have implemented it as \LaTeX2e package without disturbing the existing \LaTeX reference system.

### 1.3 Interfaces

The package provides a generic interface for programmers. Commands of this interface are prefixed by \texttt{\zref@}.

- Option \texttt{user} enables the user interface. Here the commands are prefixed by \texttt{\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 \texttt{zref-}, for example:

\begin{verbatim}
\RequirePackage{zref-apspage}
\end{verbatim}

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

As alternative package \texttt{zref} can be used and the modules are given as options:

\begin{verbatim}
\usepackage[perpage,user]{zref}
\end{verbatim}

### 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:

\begin{verbatim}
\Z@R@myref \rightarrow \default\{2.1\}\page\{7\}
\end{verbatim}

The generic format of a entry is:

\begin{verbatim}
\Z@R\langle\texttt{refname}\rangle \rightarrow \langle\texttt{propname}\rangle\{\langle\texttt{value}\rangle\}
\end{verbatim}

\texttt{\langle\texttt{refname}\rangle} is the name that denoted references (the name used in \texttt{\label} and \texttt{\ref}).

\texttt{\langle\texttt{propname}\rangle} is the name of the property or key. The property key macro is never executed, it is used in parameter text matching only.

**Property.** Because the name of a property is used in a macro name that must survive the \texttt{.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 \texttt{\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 \LaTeX’s `\currentlabel`.

\texttt{\zref@getcurrentexp}{⟨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`.

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

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

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

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

\subsection{Reference generation}

\texttt{\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.

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

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

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

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

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

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

\subsection{Data extraction}

\texttt{\zref@extractdefaultexp}{⟨refname⟩}{⟨propname⟩}{⟨default⟩}

This is the basic command that refersnce the value of a property ⟨propname⟩ for the reference ⟨refname⟩. In case of errors such as undefined reference the ⟨default⟩ is used instead.
\ref@extract\exp\{\langle refname\rangle\}\{\langle propname\rangle\}

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

Example for page references:

\LaTeX: \pageref{foobar}
\zref: \ref@extract{foobar}{page}

Both \ref@extract and \ref@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 \ref@re-
\fused and \ref@wrapper@babel for its user macros.

\ref@refused\{\langle refname\rangle\}

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

\ref@def@extract\{\langle cmd\rangle\}\{\langle refname\rangle\}\{\langle propname\rangle\}
\ref@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 \ref@extract and \ref@extractdefault. The result is stored
in macro \langle cmd\rangle. Also \ref@refused is called to notify \LaTeX\ that the reference
\langle refname\rangle is used. Added in 2011/10/04 v2.22.

\ref@ifrefundefined\exp\{\langle refname\rangle\}\{\langle then\rangle\}\{\langle else\rangle\}

Macro \ref@ifrefundefined calls arguments \langle then\rangle or \langle else\rangle dependent on the
existence of the reference \langle refname\rangle.

\ifrefundefined\{\langle refname\rangle\}\{\langle then\rangle\}\{\langle else\rangle\}

Macro \ifrefundefined calls \ref@refused before executing \ref@ifrefundefined. Babel
shorthands are supported in \langle refname\rangle.

\ref@ifrefcontainsprop\exp\{\langle refname\rangle\}\{\langle propname\rangle\}\{\langle then\rangle\}\{\langle else\rangle\}

Test whether a reference provides a property.

2.6 Setup

\ref@default

Holds the global default for unknown values.

\ref@setdefault\{\langle value\rangle\}

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.

\setmainlist \{value\}

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

### 2.7 Declared properties

<table>
<thead>
<tr>
<th>Module</th>
<th>Property</th>
<th>Property list</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{(base)}</td>
<td>default</td>
<td>main</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td></td>
<td>page</td>
<td>main</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td>abspage</td>
<td>abspage</td>
<td>main</td>
<td>0</td>
</tr>
<tr>
<td>counter</td>
<td>counter</td>
<td>main</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td>hyperref</td>
<td>anchor</td>
<td>main</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td></td>
<td>url</td>
<td>main</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td>pageattr</td>
<td>pdfpageattr</td>
<td>thepage</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>pdfpagesattr</td>
<td>LastPage</td>
<td>...</td>
</tr>
<tr>
<td>\texttt{pagelayout}(^1)</td>
<td>mag</td>
<td>thepage</td>
<td>\number\mag</td>
</tr>
<tr>
<td></td>
<td>paperwidth</td>
<td>thepage</td>
<td>\number\paperwidth</td>
</tr>
<tr>
<td></td>
<td>paperheight</td>
<td>thepage</td>
<td>\number\paperheight</td>
</tr>
<tr>
<td></td>
<td>stockwidth</td>
<td>thepage</td>
<td>\number\stockwidth</td>
</tr>
<tr>
<td></td>
<td>stockheight</td>
<td>thepage</td>
<td>\number\stockheight</td>
</tr>
<tr>
<td></td>
<td>pdfpageheight</td>
<td>thepage</td>
<td>\number\pdfpageheight</td>
</tr>
<tr>
<td></td>
<td>pdfpagewidth</td>
<td>thepage</td>
<td>\number\pdfpagewidth</td>
</tr>
<tr>
<td></td>
<td>pdfhorigin</td>
<td>thepage</td>
<td>\number\pdfhorigin</td>
</tr>
<tr>
<td></td>
<td>pdfvorigin</td>
<td>thepage</td>
<td>\number\pdfvorigin</td>
</tr>
<tr>
<td></td>
<td>hoffset</td>
<td>thepage</td>
<td>\number\hoffset</td>
</tr>
<tr>
<td></td>
<td>voffset</td>
<td>thepage</td>
<td>\number\voffset</td>
</tr>
<tr>
<td></td>
<td>totpmargin</td>
<td>thepage</td>
<td>\number\topmargin</td>
</tr>
<tr>
<td></td>
<td>oddsidemargin</td>
<td>thepage</td>
<td>\number\oddsidemargin</td>
</tr>
<tr>
<td></td>
<td>evensidemargin</td>
<td>thepage</td>
<td>\number\evensidemargin</td>
</tr>
<tr>
<td></td>
<td>textwidth</td>
<td>thepage</td>
<td>\number\textwidth</td>
</tr>
<tr>
<td></td>
<td>textheight</td>
<td>thepage</td>
<td>\number\textheight</td>
</tr>
<tr>
<td></td>
<td>headheight</td>
<td>thepage</td>
<td>\number\headheight</td>
</tr>
<tr>
<td></td>
<td>headsep</td>
<td>thepage</td>
<td>\number\headsep</td>
</tr>
<tr>
<td></td>
<td>footskip</td>
<td>thepage</td>
<td>\number\footskip</td>
</tr>
<tr>
<td></td>
<td>marginparwidth</td>
<td>thepage</td>
<td>\number\marginparwidth</td>
</tr>
<tr>
<td></td>
<td>marginparsep</td>
<td>thepage</td>
<td>\number\marginparsep</td>
</tr>
<tr>
<td></td>
<td>columnwidth</td>
<td>thepage</td>
<td>\number\columnwidth</td>
</tr>
<tr>
<td></td>
<td>columnsep</td>
<td>thepage</td>
<td>\number\columnsep</td>
</tr>
<tr>
<td></td>
<td>perpage</td>
<td>pagevalue</td>
<td>perpage</td>
</tr>
<tr>
<td></td>
<td>page</td>
<td>perpage</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td></td>
<td>abspage</td>
<td>perpage</td>
<td>0</td>
</tr>
<tr>
<td>savepos</td>
<td>posx</td>
<td>savepos</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>posy</td>
<td>savepos</td>
<td>0</td>
</tr>
<tr>
<td>titleref</td>
<td>title</td>
<td>main</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td>\texttt{xr}</td>
<td>anchor</td>
<td>&lt;empty&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>externaldocument</td>
<td>&lt;empty&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>theotype</td>
<td>&lt;empty&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>title</td>
<td>&lt;empty&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>url</td>
<td>&lt;empty&gt;</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\)Module \texttt{pagelayout} only defines properties if the parameter exists.
2.8 Wrapper for advanced situations

\zref@wrapper@babel \{\ldots\} \{\langle \text{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 \{\ldots\}

There are situations where a label must be written instantly to the .aux file, for example after the last page. If the \label 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 \{\ldots\}

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 ɛ-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.

\the\zref@unique

This command is used to generate unique label names.

3 User interface

3.1 Module user

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

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

Babel shorthands are allowed.

robust

Robust macro.

exp

Expandable version:
• robust, unless the extracted values are fragile,
• no babel shorthand support.

exp2

Expandable like \texttt{exp} and:
• expandable in exact two steps.

The basic user interface of the package without modules are commands that mimic the standard \LaTeX{} behaviour of \texttt{\textbackslash label}, \texttt{\textbackslash ref}, and \texttt{\textbackslash pageref}:

\texttt{\zlabel{⟨refname⟩}} babel

Similar to \texttt{\textbackslash label}. It generates a label with name \texttt{⟨refname⟩} in the new reference scheme.

\texttt{\zref{⟨propname⟩}{⟨refname⟩}} babel

Without optional argument similar to \texttt{\textbackslash ref}, it returns the default reference property. This property is named \texttt{default}:

\texttt{\zref{x} ≡ \zref{default}{x}}

\texttt{\zpageref{⟨refname⟩}} babel

Convenience macro, similar to \texttt{\textbackslash pageref}.

\texttt{\zpageref{x} ≡ \zref{page}{x}}

\texttt{\zrefused{⟨refname⟩}} babel

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

3.2 Module \texttt{abspage}

With the help of package \texttt{atbegshi} a new counter \texttt{abspage} with absolute page numbers is provided. Also a new property \texttt{abspage} is defined and added to the main property list. Thus you can reference the absolute page number:

Section \texttt{\zref{foo}} is on page \texttt{\zpageref{foo}}.
This is page \texttt{\zref{abspage}{foo}}
of \texttt{\zref{abspage}{LastPage}}.

The example also makes use of module \texttt{lastpage}.

3.3 Module \texttt{lastpage}

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

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

or
\zpageref{LastPage} (module user)

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

3.3.1 Tests for last page

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

\zref@iflastpage \(\langle \text{refname} \rangle \) \{\langle \text{then} \rangle \} \{\langle \text{else} \rangle \}

Macro \zref@iflastpage compares the references \(\langle \text{refname} \rangle \) with \(\langle \text{LastPage} \rangle \). 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 \(\langle \text{then} \rangle \) is executed, otherwise code \(\langle \text{else} \rangle \) is called. If one or both references are undefined or lack the property abspage, then \(\langle \text{else} \rangle \) is executed.

Macro \zref@iflastpage is expandable, therefore \zref@refused should be called on \(\langle \text{refname} \rangle \) and \(\langle \text{LastPage} \rangle \).

\ziflastpage \(\langle \text{refname} \rangle \) \{\langle \text{then} \rangle \} \{\langle \text{else} \rangle \}

Macro \ziflastpage has the same function as \zref@iflastpage, but adds support for babel shorthands in \(\langle \text{refname} \rangle \) and calls \zref@refused. However macro \ziflastpage is not expandable.

3.3.2 Example

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}
\documentclass{report}
\newcounter{foo}
\renewcommand*{\thefoo}{\Alph{foo}}
\usepackage{zref-lastpage,zref-user}[2016/05/21]
\makeatletter
\zref@newprop{thefoo}{\thefoo}
\zref@newprop{valuefoo}{\the\value{foo}}
\zref@newprop{chapter}{\thechapter}
\zref@addprops{LastPage}{thefoo,valuefoo,chapter}
\makeatother
\newcommand*{\foo}{\stepcounter{foo} [Current foo: \thefoo]}
\begin{document}
\chapter{First chapter}
Last page is \zref{LastPage}.\ Last chapter is \zref[chapter]{LastPage}.\ Last foo is \zref[thefoo]{LastPage}.\ Last value of foo is \zref[valuefoo]{LastPage}.
\end{document}
\end{verbatim}
3.4 Module \texttt{thepage}

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

\begin{verbatim}
3.4 Module \texttt{thepage}
\end{verbatim}

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

\texttt{\zthepage\{⟨absolute page number⟩\}}

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

\texttt{\zref@thepage@name\exp\{⟨absolute page number⟩\}}

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

\texttt{\zref@thepage\exp\{⟨absolute page number⟩\}}
\texttt{\zref@thepage@refused\{⟨absolute page number⟩\}}

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

3.5 Module \texttt{nextpage}

\begin{verbatim}
3.5 Module \texttt{nextpage}
\end{verbatim}

Macro \texttt{\znextpage} prints \texttt{\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 \texttt{\zunknownnextpagename} is used instead. The default for this macro is the default of property \texttt{page}.
2. This page is the last page. Then \texttt{\znnextpagename} is used. Its default is empty.
3. The next page is known, then \texttt{\thepage} of the next page is used (the value of property \texttt{page} of the next page).

3.5.1 Configuration

The behaviour can be configured by the following macros.

\texttt{\znnextpage}
If the next page is not known or available, then \znextpage uses these name macros as default. \zunknownnextpagename is used in case of undefined references. Default is the value of property page of the next page (\thepage). Module thepage is used. Macro \znonextpagename is used, if the next page does not exists. That means that the current page is last page. The default is empty.

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

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

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

3.5.2 Example

\begin{document}
\begin{example}
37 (*example-nextpage)
38 \%<<END_EXAMPLE
39 \documentclass{book}
40 \usepackage{zref-nextpage}[2016/05/21]
41 \znextpagesetup
42 {\thepage}% next page is unknown
43 {\thepage\ (#1)}% this page is last page
44 {\thepage\ $\rightarrow$ #1}% next page is known
45 \renewcommand*{\znonextpagename}{last page}
46 \usepackage{fancyhdr}
47 \pagestyle{fancy}
48 \fancyhf{}
49 \fancyhead[LE,RO]{\znextpage}
50 \fancypagestyle{plain}{%
51 \fancyhf{}%
52 \fancyhead[LE,RO]{\znextpage}%
53 }
54 \begin{document}
55 \frontmatter
56 \tableofcontents
57 \mainmatter
58 \chapter{Hello World}
59 \clearpage
60 \section{Last section}
61 \end{document}
62 \%END_EXAMPLE
63 \end{example}
\end{document}

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.
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 (\texttt{\numexpr}) or counter assignments.

3.7 Module \texttt{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 \texttt{memoir}) or the \TeX engine like \texttt{pdf\TeX}. 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 \texttt{thepage} is loaded that generates a label for each page. The properties of module \texttt{pagelayout} are added to the property list \texttt{thepage} of module \texttt{thepage}.

List of properties:

<table>
<thead>
<tr>
<th>parameter</th>
<th>property</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>\mag</td>
<td>mag</td>
<td></td>
</tr>
<tr>
<td>\paperwidth</td>
<td>paperwidth</td>
<td></td>
</tr>
<tr>
<td>\paperheight</td>
<td>paperheight</td>
<td></td>
</tr>
<tr>
<td>\stockwidth</td>
<td>stockwidth</td>
<td>class memoir</td>
</tr>
<tr>
<td>\stockheight</td>
<td>stockheight</td>
<td>class memoir</td>
</tr>
<tr>
<td>\pdfpagewidth</td>
<td>pdfpagewidth</td>
<td>\texttt{pdf\TeX}, \texttt{Lua\TeX}</td>
</tr>
<tr>
<td>\pdfpageheight</td>
<td>pdfpageheight</td>
<td>\texttt{pdf\TeX}, \texttt{Lua\TeX}</td>
</tr>
<tr>
<td>\pdfhorigin</td>
<td>pdfhorigin</td>
<td>\texttt{pdf\TeX}, \texttt{Lua\TeX}</td>
</tr>
<tr>
<td>\pdfvorigin</td>
<td>pdfvorigin</td>
<td>\texttt{pdf\TeX}, \texttt{Lua\TeX}</td>
</tr>
<tr>
<td>\hoffset</td>
<td>hoffset</td>
<td></td>
</tr>
<tr>
<td>\voffset</td>
<td>voffset</td>
<td></td>
</tr>
<tr>
<td>\topmargin</td>
<td>topmargin</td>
<td></td>
</tr>
<tr>
<td>\oddsidemargin</td>
<td>oddsidemargin</td>
<td></td>
</tr>
<tr>
<td>\evensidemargin</td>
<td>evensidemargin</td>
<td></td>
</tr>
<tr>
<td>\textwidth</td>
<td>textwidth</td>
<td></td>
</tr>
<tr>
<td>\textheight</td>
<td>textheight</td>
<td></td>
</tr>
<tr>
<td>\headheight</td>
<td>headheight</td>
<td></td>
</tr>
<tr>
<td>\headsep</td>
<td>headsep</td>
<td></td>
</tr>
<tr>
<td>\footskip</td>
<td>footskip</td>
<td></td>
</tr>
<tr>
<td>\marginparwidth</td>
<td>marginparwidth</td>
<td></td>
</tr>
<tr>
<td>\marginparsep</td>
<td>marginparsep</td>
<td></td>
</tr>
<tr>
<td>\columnwidth</td>
<td>columnwidth</td>
<td></td>
</tr>
<tr>
<td>\columnsep</td>
<td>columnsep</td>
<td></td>
</tr>
</tbody>
</table>

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

3.8 Module \texttt{marks}

ToDo.
3.9 Module runs

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

\texttt{\textbackslash runsexp}

Prints the the total number of \LaTeX{} runs including the current one. It expands to an explicit number. Before \texttt{\textbackslash 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 \texttt{\textbackslash runs} is set to one. \LaTeX{} runs where the .aux files are not rewritten are not counted (see \texttt{\textbackslash nofiles}).

3.10 Module perpage

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

\texttt{\textbackslash makeperpage\{(reset)\}\{\textbackslash \{counter\}\}}

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

\texttt{\textbackslash makeperpage\{footnote\} \% \usepackage\{perpage\}\{footmisc\}}

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

\texttt{\textbackslash makeperpage\{2\}\{footnote\}}

\texttt{\textbackslash 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 \texttt{\textbackslash thezpage}, the page number of the current page. Or counter \texttt{zpage} can be used, if the page number should be formatted differently from the current page number. Example:

\texttt{\newcounter\{foobar\}}
\texttt{\makeperpage\{foobar\}}
\texttt{\renewcommand\{\thefoobar\}\{\thezpage\-\arabic\{foobar\}}}
\% or
\texttt{\renewcommand\{\thefoobar\}\{\roman\{zpage\}\-\arabic\{foobar\}}

\texttt{\textbackslash unmakeperpage\{\textbackslash \{counter\}\}}

The reset mechanism for this counter is deactivated.

3.11 Module counter

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

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

\titleref\{⟨refname⟩\}

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

\titlerefsetup \{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 XƎTeX). 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.

\savepos\{⟨refname⟩\}

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

\saveposx\{⟨refname⟩\}
\saveposy\{⟨refname⟩\}

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

\posx\{⟨refname⟩\}
\posy\{⟨refname⟩\}

Get the position as number. Unit is sp. Horizontal positions by \posx increase from left to right. Vertical positions by \posy 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 \texttt{zposx} and \texttt{zposy} are expandable and can be used inside calculations (\texttt{setcounter, addtocounter}, package \texttt{calc}, \texttt{numexpr}). However this property prevents from notifying \LaTeX{} that the reference is actually used (the notifying is not expandable). Therefore you should mark the reference as used by \texttt{zrefused}.

This module uses pdfTeX’s \texttt{pdfsavepos}, \texttt{pdflastxpos}, and \texttt{pdflastypos}. They are available in PDF mode and since version 1.40.0 also in DVI mode.

### \texttt{zref\@savepos}

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

Thus \texttt{zsavepos} is basically \texttt{zref\@savepos} followed by \texttt{zref\@label-bylist('refname')}\{\texttt{savepos}\}. If \TeX{}\texttt{XeTstate} is detected and enabled, \texttt{savepos} also adds \texttt{zref\@savepos} at the end to support \texttt{beginR} where the whatits are processed in reverse order. The property list \texttt{savepos} contains the properties \texttt{posx} and \texttt{posy}.

### 3.14 Module \texttt{dotfill}

### \texttt{zdotfill}

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

### \texttt{zdotfill\@setup\{key\textsubscript{1}=value\textsubscript{1}, key\textsubscript{2}=value\textsubscript{2}, \ldots\}}

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

- \texttt{min=\{count value\}}
  - If the actual number of dots are smaller than \texttt{\{count value\}}, then the dots are suppressed. Default: 2.

- \texttt{unit=\{dimen value\}}
  - The width of a dot unit is given by \texttt{\{dimen value\}}. Default: \texttt{0.44em} (same as the unit in \texttt{dotfill}).

- \texttt{dot=\{value\}}
  - The dot itself is given by \texttt{\{value\}}. Default: \texttt{.} (dot, same as the dot in \texttt{dotfill}).

### 3.15 Module \texttt{env}

This module defines two properties \texttt{envname} and \texttt{envline}. They remember the name of the environment and the line number at the start of the environment.

### 3.16 Module \texttt{xr}

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

### \texttt{externaldocument* \{\langle prefix\rangle\textsubscript{\texttt{prefix}} \{\langle external document\rangle\} \{\langle url\rangle\}}

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

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

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

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

\zxrsetup {key_1=value_1, key_2=value_2, ...}

The following setup options are available:

\textbf{ext}: It sets the default extension.

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

\textbf{toltxlabel}: Boolean option. The found references are imported as L\TeX labels.
Packages nameref, titleref and class memoir are supported.

\textbf{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.

\textbf{verbose}: Boolean option. List the imported labels in the \log file. Default is false.

\zref@xr@ext

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

4 ToDo

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

\begin{itemize}
  \item Other applications: autoref, hyperref, ...
\end{itemize}

5 Example

\begin{verbatim}
67 (*example)
68 \documentclass{book}
69 \usepackage[ngerman]{babel}%
70 \usepackage{nameref, titleref, dotfill, counter, user}{zref}
71 72 \zref@wrapper@immediate

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.
\end{verbatim}

19
Also this example shows how chapter titles can be recorded. A new property `chaptitle` is declared and added to the main property list. In \texttt{\ChapterStart} the current value of the property is updated.

\begin{verbatim}
\newcommand*{\ChapterStart}[2]{\% 
  \zrefreset{chaptitle}{} \makeatletter \zref@newprop{chaptitle}{} \zref@addprop{main}{chaptitle} \makeatother \\cleardoublepage \def\current@chapid{#1} \zref@setcurrent{chaptitle}{#2} \chapter{#2} \zlabel{chap:#1} \}
\newcommand*{\ChapterStop}{\% \cleardoublepage \zref@wrapper@immediate{\% \zref@labelbyprops{chapend:\current@chapid}{abspage}\% }\% \makeatother \begin{document} \end{verbatim}

\texttt{\ChapterPages} calculates and returns the number of pages of the referenced chapter.

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

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

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

\tableofcontents

\mainmatter

\ChapterStart{first}{First chapter}

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

\section{Test}
\zlabel{a"o}

Section \zref{a"o} on page \zref@wrapper@babel\zref@extract{a"o}{page}.

Text.
\newpage

\section{Hello World}
\zlabel{hello}

\ChapterStop

Here an example follows that makes use of pdf\TeX{}’s “savepos” feature. The position on the page is not known before the page is constructed and shipped out. Therefore the position ist stored in references and are available for calculations in the next \TeX{} compile run.

The width of the first column is
\the\dimexpr \zposx{secondcol}sp - \zposx{firstcol}sp\relax,\$
\the\dimexpr \zposy{firstcol}sp - \zposy{secondline}sp\relax$:
\begin{tabular}{ll}
Hello & World \\
Second line & foobar
\end{tabular}

With \zrefused \TeX{} is notified, if the references are not yet available and \TeX{} can generate the rerun hint.

\zrefused{firstcol}
\zrefused{secondcol}
\zrefused{secondline}
\ChapterStop

Test for module \dotfill.

\newcommand*{\dftest}[1]{%
#1&$[$\verb|\dotfill|$]$&$[$\verb|\zdotfill|$]$\\%
}[1]{%
#1&$[$\verb|\dotfill|$]$&$[$\verb|\zdotfill|$]$\\%
}
6 Implementation

6.1 Package zref

6.1.1 Identification

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref}[2016/05/21 v2.26 A new reference scheme for LaTeX (HO)]

6.1.2 Load basic module

Abort package loading if zref-base could not be loaded successfully.
\@ifundefined{ZREF@base@ok}{\endinput}{}

6.1.3 Process options

Known modules are loaded and the release date is checked.
\def\ZREF@temp#1{\DeclareOption{#1}{\AtEndOfPackage{\RequirePackage{zref-#1}[2016/05/21]}}}
\ProcessOptions

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:
\zref@: Macros of the programmer’s interface.
\ZREF@: Internal macros.
\Z@L@listname: The properties of the list \langle listname \rangle.
\Z@D@propname: The default value for property \langle propname \rangle.
\Z@E@propname: Extract function for property \langle propname \rangle.
\Z@X@propname: Information whether a property value for property \langle propname \rangle is expanded immediately or at shipout time.
\Z@C@propname: Current value of the property \langle propname \rangle.
\Z@R@labelname: Data for reference \langle labelname \rangle.
\ZREF@org@: Original versions of patched commands.
\z: For macros in user land, defined if module user is set.

The following family names are used for keys defined according to the keyval package:
\ZREF@TR: Setup for module titleref.

6.2.2 Identification

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-base}[2016/05/21 v2.26 Module base for zref (HO)]

6.2.3 Utilities

\RequirePackage{ltxcmds}[2010/12/02]
\RequirePackage{infwarerr}[2010/04/08]
\RequirePackage{kvsetkeys}[2010/03/01]
\RequirePackage{kvdefinekeys}[2010/03/01]
\RequirePackage{pdftexcmds}[2010/04/01]

\ZREF@name
Several times the package name is used, thus we store it in \ZREF@name.
\def\ZREF@name{zref}
\ltx@IfUndefined{protected}{% 
\RequirePackage{makerobust}[2006/03/18]%

\ZREF@Robust
\def\ZREF@Robust#1#2{% 
\def\ZREF@Robust#1{% 
\protected#1% 
)}% 
}

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

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

23
\ZREF@UpdatePdfTeX \ZREF@UpdatePdfTeX is used as help message text in error messages.
240 \def\ZREF@UpdatePdfTeX{Update pdfTeX}

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

\ZREF@patch Macro \ZREF@patch first checks the existence of the command and saves it.
248 \def\ZREF@patch#1{%
249 \ltx@ifUndefined{#1}{% 
250 \ltx@gobble 
251 }{% 
252 \expandafter\let\csname ZREF@org@#1\expandafter\endcsname 
253 \csname #1\endcsname 
254 \ltx@firstofone 
255 }%
256 }

6.2.4 Check for \textit{\eTeX} 

The use of \textit{\eTeX} should be standard nowadays for \LaTeX. We test for \textit{\eTeX} in order to use its features later.
257 \ltx@ifUndefined{eTeXversion}{%
258 \PackageError\ZREF@name{Missing support for \textit{\eTeX}; package is abandoned}{%
259 \PackageError\ZREF@name{Use a TeX compiler that support \textit{\eTeX} and enable \textit{\eTeX} in the format}{%
260 }
261 Use a TeX compiler that support \textit{\eTeX} and enable \textit{\eTeX} %
262 in the format. %
263 }
264 \endinput
265 }
266 \RequirePackage{etexcmds}[2007/09/09]
267 \ifetex@unexpanded
268 \else
269 \PackageError\ZREF@name{Missing \textit{\eTeX}'s \string\unexpanded.\MessageBreak
270 Add \string\RequirePackage\string{etexcmds}\string{etexcmds}\string} before %
271 \string\documentclass%
272}
273 }
274 Probably you are using some package (e.g. ConTeXt) that %
275 redefines \string\unexpanded%
276 }
277 \expandafter\endinput
278 }

6.2.5 Auxiliary file stuff

We are using some commands in the .aux files. However sometimes these auxiliary files are interpreted by \LaTeX processes that haven’t loaded this package (e.g. package \texttt{xr}). Therefore we provide dummy definitions.
279 \RequirePackage{auxhook}
280 \AddLineBeginAux{%
281 \string\providecommand\string{zref@newlabel}[2]{%}
282 }

\ZREF@RefPrefix
283 \def\ZREF@RefPrefix{Z@R}

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

- \Z@R@labelname is defined.
• \LaTeX{}’s check for multiple references.
• \LaTeX{}’s check for changed references.

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

6.2.6 Property lists

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

287 \ZREF@Robust\def\zref@newlist#1{%
288 \zref@iflistundefined{#1}{%
289 \ltx@secondoftwo
290 }{%
291 \PackageInfo\ZREF@name{New property list: #1}%
292 )%
293 }%
294 }%
295 }%
296 }%;
297 }%
298 }

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

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

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

302 \ZREF@Robust\def\zref@listexists#1{%
303 \zref@iflistundefined{#1}{%
304 \PackageError\ZREF@name{%
305 Property list `#1' does not exist%
306 })%
307 }%
308 }

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

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

\zref@listforloop

320 \def\zref@listforloop#1{%
321 \zref@listexists{#1}{%
322 \zref@iflistcontainsprop{#1}{}%
323 }%
324 }%
\zref@addprops \zref@addprop adds the properties \#2 to the property list \#1, if the property is not already in the list. Otherwise a warning is given.
\ZREF@Robust\def\zref@addprops#1#2{%\zref@listexists{#1}{%\comma@parse{#2}{%\zref@propexists\comma@entry{%\zref@iflistcontainsprop{#1}{\comma@entry}{%\PackageWarning\ZREF@name{\Property `\comma@entry' is already in list `\#1'\}}{\begingroup\expandafter\endgroup\expandafter\g@addto@macro\csname Z@L@#1\expandafter\endcsname\expandafter{\csname\comma@entry\endcsname}}}}}{\ltx@gobble}{%}}%\zref@localaddprops \zref@localaddprops adds the properties \#2 to the property list \#1, if the property is not already in the list. Otherwise a warning is given.
\ZREF@Robust\def\zref@localaddprops#1#2{%\zref@listexists{#1}{%\comma@parse{#2}{%\zref@propexists\comma@entry{%\zref@iflistcontainsprop{#1}{\comma@entry}{%\PackageWarning\ZREF@name{\Property `\comma@entry' is already in list `\#1'\}}{\begingroup\expandafter\endgroup\expandafter\g@addto@macro\csname Z@L@#1\expandafter\endcsname\expandafter{\csname#2\endcsname}}}}}{\ltx@gobble}{%}}%
\ZREF@delprop
\def\ZREF@delprop#1{%
\expandafter\ZREF@@delprop\expandafter{\string#1}#1%
}%

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

\zref@delprop
\ZREF@Robust\def\zref@delprop{%
\ZREF@delprop\xdef
}\zref@delprop
\ZREF@Robust\def\zref@delprop{%
\ZREF@delprop\edef
}\zref@delprop
\ZREF@delprop
\def\ZREF@delprop#1#2#3{%
\zref@listexists{#2}{%
\def\ZREF@param{#3}
\edef\ZREF@SavedEscapechar{\the\escapechar}%
\escapechar=-1%
\expandafter#1\csname Z@L@#2\endcsname!%
\escapechar=\ZREF@SavedEscapechar\relax%
}%
\zref@delprop
Caution: #1 might be an \lf or similar token.
\def\ZREF@delprop#1{%
\expandafter\ZREF@@delprop\expandafter{\string#1}#1%
}%

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

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

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

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

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

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

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

\Z@E@propname: Extract function.

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

\Z@C@propname: Current value of the property.
\zref@setcurrent \zref@setcurrent sets the current value for a property.

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

\ZREF@getcurrent \zref@getcurrent gets the current value for a property.
\def\ZREF@getcurrent#1{\%
  \romannumeral0\%
  \ltx@ifundefined{Z@C@#1}{\%
    \ltx@space\%
  }{\%
    \expandafter\expandafter\expandafter\ltx@space\%
    \csname Z@C@#1\endcsname\%
  }\%
}\%

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

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

6.2.8 Reference generation

\zref@label Label macro that uses the main property list.
\ZREF@Robust\def\zref@label#1{\%
  \zref@labelbylist{#1}\ZREF@mainlist\%
}\%

\zref@labelbylist Label macro that stores the properties, specified in the property list \#2.
\ZREF@Robust\def\zref@labelbylist#1#2{\%
  @bsphack\%
  \zref@listexists{#2}{\%
    \expandafter\expandafter\expandafter\zref@label\%
    \expandafter\expandafter\expandafter{\%
      \csname Z@L@#2\endcsname{#1}\%
    }\%
  }\%
  \@esphack\%
}\%

\zref@labelbyprops The properties are directly specified in a comma separated list.
\ZREF@Robust\def\zref@labelbyprops#1#2{\%
  @bsphack\%
  \begingroup\%
  \toks@{}\%
  \comma@parse{#2}{\%
    \zref@ifpropundefined\comma@entry{\%
      \PackageWarning\ZREF@name{\%
        Property \textbackslash\comma@entry is not known\%
      }\%
    }\%
    \comma@parse{#1\%
  }\%
  \@esphack\%
}\%
\%

31
Option `immediate' expects `true' or `false'.
Ignoring invalid value `\ZREF@param'\%
}
\fi
\fi
\ZREF@false
\def\ZREF@false{false}
\ZREF@true
\def\ZREF@true{true}
\kv@define@key{ZREF@LABEL}{values}{%
\kv@parse{#1}{%
\ifx\kv@value\relax
\@PackageWarning\ZREF@name{%
Missing value for property `\kv@key'%
}%
\expandafter\ltx@gobbletwo
\else
\expandafter\zref@setcurrent
\fi
\}%
}%
\ifZREF@immediate
The switch \ifZREF@immediate tells us, whether the label should be written immediately or at page shipout time. \ZREF@label need to be notified about this, because it must disable the deferred execution of property values, if the label is written immediately.
\newif\ifZREF@immediate
\zref@wrapper@immediate
The argument of \zref@wrapper@immediate is executed inside a group where \write is redefined by adding \immediate before its execution. Also \ZREF@label is notified via the switch \ifZREF@immediate.
\ZREF@label writes the data in the .aux file. #1 contains the list of valid properties, #2 the name of the reference. In case of immediate writing, the deferred execution of property values is disabled. Also 33is made expandable in this case.
Design goal for the extracting macros is that the extraction process is full expandable. Thus these macros can be used in expandable contexts. But there are problems that cannot be solved by full expandable macros:

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

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

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

\zref@ifrefundefined\texttt{#1}\%\zifrefundefined\texttt{#2}\texttt{#3}\%\zifrefundefined\texttt{#1}\%\zifrefundefined
The problem with undefined references is addressed by the macro \zref@refused. This can be used outside the expandable context. In case of an undefined reference the flag is set to notify \LaTeX{} and a warning is given.

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

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

\zref@extract \zref@extract is an abbreviation for the case that the default of the property is used as default value.
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.

\ZREF@extractdefault

\ZREF@wu@extractdefault

\ZREF@extractdefault

\ZREF@def@extract

\ZREF@def@extract

\ZREF@def@extract
6.2.10 Compatibility with babel
6.2.11 Unique counter support

\zref@require@unique

Generate the counter \zref@unique if the counter does not already exist.

\ZREF@Robust\def\zref@require@unique{%
\@ifundefined{c@zref@unique}{%
\begingroup
\let\@addtoreset\ltx@gobbletwo
\newcounter{zref@unique}%
\endgroup
\thezref@unique
\thezref@unique
\zref@number\c@zref@unique
}%
\relax}{}

thezref@unique

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

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

6.2.12 Utilities

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

6.2.13 Setup

\zref@setdefault

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

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

\zref@default

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

\zref@setdefault{%
\reset@font\bfseries ??\%%
}

Main property list.

\zref@setmainlist

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

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

Now we create the list.

\zref@newlist\ZREF@mainlist
Main properties. The two properties `default` and `page` are created and added to the main property list. They store the data that standard \LaTeX uses in its references created by `\label`.

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

`page` the appearance of the page counter

Properties

\ZREF@NewPropAnchor

\ZREF@NewPropTitle

\ZREF@NewPropTheotype

\ZREF@NewPropPageValue

Mark successful loading

6.3 Module user

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

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

\zkvlabel
\newcommand*{\zkvlabel}[1]{\ifx\label\ltx@gobble\expandafter\ltx@gobblethree\fi\zref@wrapper@babel{\zref@labelbykv{#1}}}\zpageref
For macro \zpageref, we just call \zref with property page.
\ZREF@IfDefinable\zpageref\def{\zref[page]}\zrefused
For the following expandible user macros \zrefused should be used to notify \LaTeX in case of undefined references.
\ZREF@IfDefinable\zrefused\def{\zref@refused} \zrefused

6.4 Module abspage
\zpageref
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.
\zrefused
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.
Note that counter \texttt{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 \texttt{\zref@wrapper@immediate}, then the number is too small by one.

6.5 Module \texttt{counter}

For features such as hyperref’s \texttt{autoref} we need the name of the counter. The property \texttt{counter} is defined and added to the main property list.

\texttt{\zref@newprop(counter)}

\texttt{\zref@addprop\ZREF@mainlist(counter)}

\texttt{\refstepcounter} is the central macro where we know which counter is responsible for the reference.

6.6 Module \texttt{lastpage}

The module \texttt{lastpage} implements the service of package \texttt{lastpage} by setting a reference \texttt{LastPage} at the end of the document. If module \texttt{abspage} is given, also the absolute page number is available, because the properties of the main property list are used.
6.7 Module thepage
6.8 Module nextpage

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-nextpage}[2016/05/21 v2.26 Module nextpage for zref (HO)]
\RequirePackage{zref-base}[2016/05/21]
\RequirePackage{zref-abspage}[2016/05/21]
\RequirePackage{zref-thepage}[2016/05/21]
\RequirePackage{zref-lastpage}[2016/05/21]
\RequirePackage{uniquecounter}[2009/12/18]
\UniqueCounterNew{znextpage}
\newcommand*{\znextpagesetup}{\afterassignment\ZREF@np@setup@i
\afterassignment\ZREF@np@setup@ii
\afterassignment\ZREF@np@setup@iii
\afterassignment\ZREF@np@setup@iv}
\def\ZREF@np@setup@i{\afterassignment\ZREF@np@setup@ii
\def\ZREF@np@call@next##1{}}
\def\ZREF@np@setup@ii{\def\ZREF@np@call@nonext##1{}}
\def\ZREF@np@setup@iii{\def\ZREF@np@call@unknown#1{#1}}
\def\ZREF@np@setup@iv{\def\ZREF@np@call@unknown#1{}}
\zthepage
\ZREF@IfDefinable\zthepage\def{\znextpagesetup}{% #1{\zref@thepage@name{#1}{page}}\zref@thepage{#1}}%
\endinput

\ZREF@Robust\def\zref@thepage@refused#1{\zref@refused{\zref@thepage@name{#1}}}%
\edef\zref@thepage@name#1{thepage\number#1}%
\def\zref@thepage#1{\zref@extract{\zref@thepage@name{#1}}{page}{}%}
\zref@thepage@refused
\edef\zref@thepage@name#1{thepage\the\numexpr#1}%
\def\zref@thepage@name#1{thepage\number#1}%
\def\ZREF@np@call@nonext#1{#1}
\def\ZREF@np@call@next#1{#1}
\ZREF@ifDefinable\znextpage\def{%}
{\UniqueCounterCall{znextpage}\{\ZREF@nextpage\}}%
}%
\newcommand*{\znonextpagename}{}
\newcommand*{\zunknownnextpagename}{\Z@D@page}
\def\ZREF@nextpage#1{%
\begingroup
\def\ZREF@refname@this{zref@np#1}%
\zref@labelbyprops\ZREF@refname@this{abspage}%
\chardef\ZREF@call=0 % unknown
\ZREF@ifrefundefined\ZREF@refname@this{%
}%
\edef\ZREF@pagenum@this{%
\zref@extractdefault\ZREF@refname@this{abspage}{0}%
}%
\edef\ZREF@refname@next{%
\zref@thepage@name{\
\the\numexpr\ZREF@pagenum@this+1%}
}%
\ifnum\ZREF@pagenum@this>0 %
\ZREF@ifrefundefined{LastPage}{%
\zref@ifrefundefined\ZREF@refname@next{%
}\chardef\ZREF@call=2 % next page
}%
\else
\edef\ZREF@pagenum@last{%
\zref@extractdefault{LastPage}{abspage}{0}%
}%
\ifnum\ZREF@pagenum@this<\ZREF@pagenum@last \ltx@space%
\ZREF@ifrefundefined\ZREF@refname@next{%
}\chardef\ZREF@call=2 % next page
\else
\ifnum\ZREF@pagenum@this=\ZREF@pagenum@this \ltx@space%
\chardef\ZREF@call=1 % no next page
\fi
\fi
\fi
\fi
\edef\x{%
\endgroup
\ife\ZREF@call%
\noexpand\ZREF@np@call@unknown{%
\noexpand\zunknownnextpagename
}\or
\noexpand\ZREF@np@call@nonext{%
\noexpand\znonextpagename
}\else
\noexpand\ZREF@np@call@next{%
\noexpand\zref@extract{\ZREF@refname@next}{page}%
}\fi
\fi
\x
6.9 Module totpages

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-totpages}[2016/05/21 v2.26 Module totpages for zref (HO)]
\RequirePackage{zref-base}[2016/05/21]
\ifx\ZREF@base@ok Y\
\else\expandafter\endinput\fi
\RequirePackage{zref-abspage}[2016/05/21]
\RequirePackage{zref-lastpage}[2016/05/21]
\ztotpages

Macro \ztotpages contains the number of pages. It can be used inside expandable calculations. It expands to zero if the reference is not yet available.
\newcommand*{\ztotpages}{\zref@extractdefault{LastPage}{abspage}{0}}

Also we mark the reference LastPage as used:
\AtBeginDocument{\zref@refused{LastPage}}
\ztotpages

6.10 Module pagelayout

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-pagelayout}[2016/05/21 v2.26 Module pagelayout for zref (HO)]
\RequirePackage{zref-base}[2016/05/21]
\ifx\ZREF@base@ok Y\
\else\expandafter\endinput\fi
\RequirePackage{zref-thepage}[2016/05/21]
\RequirePackage{ifluatex}[2010/03/01]
\RequirePackage{atveryend}[2010/03/24]
\ifluatex
\ifnum\luatexversion<39 %
\def\ZREF@temp#1{\let#1\ltx@undefined\directlua{if tex.enableprimitives then tex.enableprimitives('', {'\string#1'}) end}}\fi
\zref@refused{LastPage}\fi
\ztotpages

6.10.1 Support for Lua\TeX
\ifluatex
\ifnum\luatexversion<39 %
\else
\begingroup
\escapechar=-1 %
\def\ZREF@temp#1{\ltx@IfUndefined{\string#1}{%\ltx@undefined\directlua{if tex.enableprimitives then tex.enableprimitives('', {'\string#1'})%end}}%}
\zref@refused{LastPage}\fi
\endgroup
\fi
\ztotpages
\global\#1=#1
6.10.2 Define layout properties

\def\ZREF@temp#1{\begingroup\escapechar=-1
\ltx@ifundefined{\string#1}{\endgroup}{\edef\x{\endgroup\noexpand\zref@newprop*{\string#1}\[
\noexpand\number\noexpand#1]\% hash-ok
\noexpand\zref@addprop{thepage}{\string#1}]}\x\endgroup}

\ZREF@temp\mag\ZREF@temp\paperwidth\ZREF@temp\paperheight\ZREF@temp\stockwidth % memoir.cls, crop.sty\ZREF@temp\stockheight % memoir.cls, crop.sty\ZREF@temp\mediawidth % VTeX\ZREF@temp\mediaheight % VTeX\ZREF@temp\pdfpagewidth\ZREF@temp\pdfpageheight\ZREF@temp\pdfhorigin\ZREF@temp\pdfvorigin\ZREF@temp\hoffset\ZREF@temp\voffset\ZREF@temp\topmargin\ZREF@temp\oddsidemargin\ZREF@temp\evensidemargin\ZREF@temp\textwidth\ZREF@temp\textheight\ZREF@temp\headheight\ZREF@temp\headsep\ZREF@temp\footskip\ZREF@temp\marginparwidth\ZREF@temp\marginparheight\ZREF@temp\marginparsep\ZREF@temp\columnwidth\ZREF@temp\columnsep\ZREF@temp\trimedge % memoir.cls\ZREF@temp\spinemargin % memoir.cls\ZREF@temp\foremargin % memoir.cls\ZREF@temp\trimtop % memoir.cls\ZREF@temp\uppermargin % memoir.cls\ZREF@temp\headmargin % memoir.cls\zref@newprop**{outputboxwd}[0pt]{\AtBeginShipoutBoxWidth}\
zref@newprop**{outputboxht}[0pt]{\AtBeginShipoutBoxHeight}
6.11 Module pageattr
\let\zref@unhex\ltx@firstofone
\let\zref@hex\pdf@escapehex
\let\zref@unhex\pdf@unescapehex
\fi
\ifZREF@pa@list
\ltx@newif\ifZREF@pa@list
\zref@listpageattr
\ZREF@IfDefinable\zlistpageattr\def{\ZREF@pa@listtrue}
\ZREF@pa@AfterLastShipout
\def\ZREF@pa@AfterLastShipout{%\ifZREF@pa@list
\edef\ZREF@page@max{\the\value{abspage}}%
ltx@ifundefined{ZREF@org@testdef}{%\let\ZREF@org@testdef\@testdef
\def\@testdef##1##2##3{%\ZREF@org@testdef{##1}{##2}{##3}%
\def\ZREF@temp{##1}%
\ifx\ZREF@temp\ZREF@RefPrefix
\expandafter\xdef\csname##1@##2\endcsname{##3}%
\fi
\x}%}
\AtVeryEndDocument{\ZREF@pa@AtVeryEnd}%
\fi
\ZREF@pa@AtVeryEnd
\ltx@ifundefined{pdfpageattr}{%\def\ZREF@pa@AtVeryEnd{}
\def\ZREF@pa@AtVeryEnd{%
\begingroup\toks@{List of \ltx@backslashchar pdfpageattr:\MessageBreak}%
\count@=1 %
\ZREF@pa@ListPage
\edef\x{\endgroup\toks@\noexpand\PackageInfoNoLine{zref-pageattr}{%\the\toks@
}\x}%
\ZREF@pa@ListPage
\edef\x{\endgroup\toks@\noexpand\PackageInfoNoLine{zref-pageattr}{%\the\toks@
}\x}%
\ZREF@pageattr
\def\zref@pageattr#1{%\zref@unhex{%\zref@extract{thepage\ZREF@number{#1}}{pdfpageattr}}%
}\zref@pageattr@used
\ZREF@Robust\def\zref@pageattr@used#1{%\zref@refused{thepage\ZREF@number{#1}}%}
6.12 Module marks

(\*)marks\NeedsTeXFormat{LaTeX2e}\ProvidesPackage{zref-marks}[2016/05/21 v2.26 Module marks for zref (HO)]\RequirePackage{zref-base}[2016/05/21]\ifx\ZREF@base@ok Y\else\expandafter\endinput\fi\newcommand*{\zref@marks@register}[3]{\zref@unhex{\zref@extract{LastPage}{pdfpagesattr}{pdfpagesattr}}}\ZREF@Robust\def{\zref@pdfpagesattr}{\zref@unhex{\zref@extract{LastPage}{pdfpagesattr}{pdfpagesattr}}}\ZREF@Robust\def{\zref@pdfpagesattr@used}{\zref@refused{LastPage}}\ltx@LocalAppendToMacro{\ZREF@Robust\def{\zref@pdfpagesattr@used}{\zref@refused{LastPage}}}\PackageInfoNoLine{zref-pageattr}{:\MessageBreak<<\zref@pdfpagesattr>>\MessageBreak}\AfterLastShipout{\ZREF@Robust\def{\zref@pdfpagesattr@used}{\zref@refused{LastPage}}}\ltx@LocalAppendToMacro{\ZREF@Robust\def{\zref@pdfpagesattr@used}{\zref@refused{LastPage}}}}
This module does not use the label-reference-system. The reference changes with each \LaTeX run and would force a rerun warning always.

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 \texttt{abspage} is loaded.

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

\begin{verbatim}
\ZREF@NewPropPageValue
\zref@newlist{perpage}
\zref@addprops{perpage}{abspage,page,pagevalue}
\end{verbatim}

The page value, known by the reference mechanism, will be stored in counter \texttt{zpage}.

Counter \texttt{zref@unique} helps in generating unique reference names.

In order to be able to reset the counter, we hook here into \texttt{\stepcounter}. In fact two nested hooks are used to allow other packages to use the first hook at the beginning of \texttt{\stepcounter}.

\begin{verbatim}
\let\ZREF@org@stepcounter\stepcounter
\def\stepcounter#1{\ifcsname @stepcounterhook@#1\endcsname\csname @stepcounterhook@#1\endcsname\fi\ZREF@org@stepcounter{#1}}
\end{verbatim}

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

\begin{verbatim}
\ZREF@IfDefinable\zmakeperpage\def{%\begin{group}
\edef\x{\endgroup\noexpand\zmakeperpage\number\numexpr#1-1\relax}
\x}
\end{verbatim}

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

\begin{verbatim}
\ZREF@IfDefinable\zmakeperpage\def{%\begin{group}
\edef\x{\endgroup\noexpand\zmakeperpage\number\numexpr#1-1\relax}
\x}
\end{verbatim}

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

\begin{verbatim}
\def\ZREF@makeperpage@opt[\ltx@zero]{%\ifnextchar[\ZREF@@makeperpage@opt\ZREF@@makeperpage{\ltx@zero}}%
\end{verbatim}

\texttt{\ZREF@@makeperpage@opt} The heart of this module follows.
First the reference is generated.
\begin{verbatim}
global\advance\c@zref@unique\ltx@one\begingroup\expandafter\zref@labelbylist\expandafter{\thezref@unique}{perpage}\%\end{verbatim}
The \texttt{\expandafter} commands are necessary, because \texttt{\ZREF@temp} is also used inside of \texttt{\zref@labelbylist}.

The evaluation of the reference follows. If the reference is not yet known, we use the page counter as approximation.
\begin{verbatim}
\zref@ifrefundefined\thezref@unique{\%\global\c@zpage=\c@page\global\let\thezpage=\thepage\expandafter\xdef\csname ZREF@abspage@#1\endcsname{\number\c@abspage}\%}%\end{verbatim}
The reference is used to set \texttt{\thezpage} and counter \texttt{zpage}.
\begin{verbatim}
\zref@extract\thezref@unique{pagevalue}{\%\global\c@zpage=\zref@extract\thezref@unique{page}{\%\xdef\thezpage{\noexpand\zref@extract{\thezref@unique}{page}}\%\expandafter\xdef\csname ZREF@abspage@#1\endcsname{\zref@extractdefault\thezref@unique{abspage}{\number\c@abspage}}}%\end{verbatim}

Page changes are detected by a changed absolute page number.
\begin{verbatim}
\expandafter\ifx\csname ZREF@abspage@#1\expandafter\endcsname\csname ZREF@currentabspage@#1\endcsname\else\global\csname c@#1\endcsname=#2\relax\global\expandafter\let\csname ZREF@currentabspage@#1\expandafter\endcsname\csname ZREF@abspage@#1\endcsname\fi\end{verbatim}

\texttt{\zunmakeperpage} \texttt{\zunmakeperpage} cancels the effect of \texttt{\zmakeperpage}.
\begin{verbatim}
\ZREF@IfDefinable\zunmakeperpage\def{\%\global\expandafter\let\csname ZREF@perpage@#1\endcsname@#1\endcsname\%\}{}\end{verbatim}

\texttt{\zunmakeperpage}\texttt{\zunmakeperpage} Macro \texttt{\zunmakeperpage} cancels the effect of \texttt{\zmakeperpage}.
\begin{verbatim}
\ZREF@IfDefinable\zunmakeperpage\def{\%\#1{\%\global\expandafter\let\\csname ZREF@perpage@#1\endcsname@#1\endcsname\%\}{}\end{verbatim}

6.15 Module \texttt{titleref}
\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}\ProvidesPackage{zref-titleref}[2016/05/21 v2.26 Module titleref for zref (HO)]\RequirePackage{zref-base}[2016/05/21]\ifx\ZREF@base@ok Y%\else\expandafter\endinput\fi\RequirePackage{gettitlestring}[2009/12/08]\endgroup\end{verbatim}

6.15.1 Implementation
\RequirePackage{keyval}

This module makes section and caption titles available for the reference system. It uses some of the ideas of package \texttt{nameref} and \texttt{titleref}.

Now we can add the property \texttt{title} is added to the main property list.

\begin{Verbatim}
\ZREF@NewPropTitle \zref@addprop{\ZREF@mainlist}{title}\
\end{Verbatim}

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

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

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

\begin{Verbatim}
\newif\ifzref@titleref@expand
\ZREF@titleref@hook
\zref@titleref@cleanup
\zref@titleref@setcurrent
\end{Verbatim}

The hook \texttt{\ZREF@titleref@hook} allows to extend the cleanup for the expansion method. Thus unnecessary macros can be removed or dangerous commands removed. The hook is executed before the expansion of \texttt{\zref@titleref@current}.

\begin{Verbatim}
\let\ZREF@titleref@hook\ltx@empty
\zref@titleref@setcurrent
\end{Verbatim}

The hook should not be used directly, instead we provide the macro \texttt{\zref@titleref@setcurrent} to add stuff to the hook and prevents that a previous non-empty content is not discarded accidently.

\begin{Verbatim}
\ZREF@Robust\def\zref@titleref@setcurrent#1{\begin{Verbatim}
\ifzref@titleref@expand
\GetTitleStringExpand{#1}\\
\else
\GetTitleStringNonExpand{#1}\\
\fi
\edef\zref@titleref@current{\detokenize{\GetTitleStringResult}}\\
\ifzref@titleref@stripperiod
\edef\zref@titleref@current{\ZREF@stripperiod\zref@titleref@current}\\
\fi
\end{Verbatim}}
\end{Verbatim}

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

\begin{Verbatim}
\newif\ifzref@titleref@stripperiod
\zref@titleref@stripperiodtrue
\zref@titleref@setcurrent
\end{Verbatim}

Macro \texttt{\zref@titleref@setcurrent} sets a new current title stored in \texttt{\zref@titleref@current}. Some cleanup and expansion is performed that can be controlled by the previous switches.
\ZREF@stripperiod  If \ZREF@stripperiod is called, the argument consists of space tokens and tokens with catcode 12 (other), because of \v-\TeX’s \detokenize.

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.

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

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

6.15.4 Environment description

6.15.5 Class memoir
6.15.6 Class beamer

\@ifclassloaded{beamer}{%}
\ZREF@patch{beamer@section}{%}
  \long\def\beamer@section[#1]{%}
  \zref@titleref@setcurrent{#1}%
  \ZREF@org@beamer@section[#1]%}
\ZREF@patch{beamer@subsection}{%}
  \long\def\beamer@subsection[#1]{%}
  \zref@titleref@setcurrent{#1}%
  \ZREF@org@beamer@subsection[#1]%}
\ZREF@patch{beamer@subsubsection}{%}
  \long\def\beamer@subsubsection[#1]{%}
  \zref@titleref@setcurrent{#1}%
  \ZREF@org@beamer@subsubsection[#1]%}
\ZREF@patch{beamer@section}{%}
  \long\def\beamer@section[#1]{%}
  \zref@titleref@setcurrent{#1}%
  \ZREF@org@beamer@section[#1]%}
\ZREF@patch{beamer@subsection}{%}
  \long\def\beamer@subsection[#1]{%}
  \zref@titleref@setcurrent{#1}%
  \ZREF@org@beamer@subsection[#1]%}
\ZREF@patch{beamer@subsubsection}{%}
  \long\def\beamer@subsubsection[#1]{%}
  \zref@titleref@setcurrent{#1}%
  \ZREF@org@beamer@subsubsection[#1]%}
\@ifpackageloaded{titlesec}{%}
\ZREF@patch{ttl@sect@i}{%}
  \def\ttl@sect@i#1#2[#3]{%}
  \zref@titleref@setcurrent{#3}%
  \ZREF@org@ttl@sect@i{#1}{#2}{#3}%
\ZREF@patch{ttl@straight@i}{%}
  \def\ttl@straight@i#1[#2]{%}
  \def\ZREF@temp{#2}%
  \ifx\ZREF@temp\ltx@empty
    \zref@titleref@setcurrent{#3}%
  \else
    \zref@titleref@setcurrent{#2}%
  \fi
  \ZREF@org@ttl@straight@i{#1}{#2}{#3}%
}
6.15.8 Package \texttt{longtable}

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

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

6.15.9 Package \texttt{listings}

Package \texttt{listings}: support for its \texttt{\caption}.\@ifpackageloaded{listings}{%  
\ZREF@patch{lst@MakeCaption}{%  
\def\lst@MakeCaption{%  
\ifx\lst@label\ltx@empty%  \else%  \expandafter\zref@titleref@setcurrent\expandafter{\lst@@caption}%  \fi%  \ZREF@org@lst@MakeCaption%  }%  
}%  
}%

6.15.10 Theorems

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

6.16 Module \texttt{xr}

\begin{verbatim}
(*xr*)
\end{verbatim}
We declare property \texttt{url}, because this is added, if a reference is imported and has not already set this field. Or if \texttt{hyperref} is used, then this property can be asked.

\begin{verbatim}
\zref@newprop{url}{}
\zref@newprop{urluse}{}
\zref@newprop{externaldocument}{}
\end{verbatim}

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

If the URL is not specified, then assume processed file with a guessed extension. Use the setting of \texttt{hyperref} if available.

\begin{verbatim}
\providecommand*{\zref@xr@ext}{\ltx@ifundefined{XR@ext}{pdf}{\XR@ext}}
\ifZREF@xr@zreflabel
\newif\ifZREF@xr@zreflabel
\SetupKeyvalOptions{family=ZREF@XR, prefix=ZREF@xr@} 
\DeclareBoolOption[true]{tozreflabel} 
\DeclareBoolOption[false]{toltxlabel} 
\DeclareBoolOption{verbose} 
\define@key{ZREF@XR}{ext}{\def\zref@xr@{#1}} 
\DeclareBoolOption[false]{urluse} 
\zxrsetup
\end{verbatim}

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

\begin{verbatim}
\ifZREF@xr@zreflabel
\newif\ifZREF@xr@zreflabel 
\SetupKeyvalOptions{family=ZREF@XR, prefix=ZREF@xr@} 
\DeclareBoolOption[true]{tozreflabel} 
\DeclareBoolOption[false]{toltxlabel} 
\DeclareBoolOption{verbose} 
\define@key{ZREF@XR}{ext}{\def\zref@xr@{#1}} 
\DeclareBoolOption[false]{urluse} 
\zxrsetup
\end{verbatim}

\begin{verbatim}
\ZREF@xr@URL
\ZREF@xr@AddURL
\end{verbatim}

\begin{verbatim}
\ZREF@xr@URL
\end{verbatim}

\begin{verbatim}
\ZREF@xr@AddURL
\end{verbatim}
\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).

\zexternaldocument  It reads the remaining arguments. \newcommand comes in handy for the optional argument.
We follow \texttt{xr} here, \texttt{ifFileExists} offers a nicer test, but we have to open the file anyway.

\begin{verbatim}
def\ZREF@xr@checkfile{%
\openin\Inputcheck\ZREF@xr@file\relax
\ifEOF\Inputcheck
\PackageWarning{zref-xr}{File `\ZREF@xr@file' not found or empty,\MessageBreak
labels not imported}%
}%
\else
\PackageInfo{zref-xr}{Label \ifZREF@xr@zreflabel (zref) \fi
import from `\ZREF@xr@file'%
}%
\def\ZREF@xr@found{0}%
\def\ZREF@xr@ignored@empty{0}%
\def\ZREF@xr@ignored@zref{0}%
\def\ZREF@xr@ignored@ltx{0}%
\ZREF@xr@processfile
\closein\Inputcheck
\begingroup
\let\on@line\ltx@empty
\PackageInfo{zref-xr}{Statistics for `\ZREF@xr@file':\MessageBreak
\ZREF@xr@found\space label(s) found%
\ifnum\ZREF@xr@ignored@empty>0 %
,\MessageBreak
\ZREF@xr@ignored@empty\space empty label(s) ignored% 
\fi
\ifnum\ZREF@xr@ignored@zref>0 %
,\MessageBreak
\ZREF@xr@ignored@zref\space duplicated zref label(s) ignored%
\fi
\ifnum\ZREF@xr@ignored@ltx>0 %
,\MessageBreak
\ZREF@xr@ignored@ltx\space duplicated latex label(s) ignored%
\fi
\endgroup
}\if\ZREF@xr@filelist\ltx@empty
\else
\edef\ZREF@xr@file{\expandafter\ltx@car\ZREF@xr@filelist\@nil}
\edef\ZREF@xr@filelist{\expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil}
\expandafter\ZREF@xr@checkfile
\fi
}\endgroup
\fi
\else\ZREF@xr@filelist\ltx@empty
\edef\ZREF@xr@filelist{%
\edef\ZREF@xr@filelist{\expandafter\ltx@car\ZREF@xr@filelist\@nil}
}\edef\ZREF@xr@filelist{%
\edef\ZREF@xr@filelist{\expandafter\ltx@cdr\ZREF@xr@filelist\@nil}
}\expandafter\ZREF@xr@checkfile
\fi
}\endgroup
\ZREF@xr@processfile
\def\ZREF@xr@processfile{%
The most work must be done for analyzing the arguments of \newlabel.
6.17 Module hyperref

UNFINISHED :-(

6.18 Module savepos

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

6.18.1 Identification

LuaTeX compatibility

68
6.18.2 Availability

First we check, whether the feature is available.

\IfUndefined{pdfsavepos}{\PackageError{ZREF@name}{pdfsavepos is not supported.\MessageBreak It is provided by pdfTeX (1.40) or XeTeX\ZREF@UpdatePdfTeX\endinput}{}}{}

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 Xe\TeX also provides \pdfsavepos.

\RequirePackage{ifpdf}
\ifpdf\else
  \IfUndefined{pdftexversion}{\ifnum\pdftexversion<140 \PackageError{ZREF@name}{pdfsavepos is not supported in DVI mode of this pdfTeX version\ZREF@UpdatePdfTeX\expandafter\expandafter\expandafter\endinput\fi}{}}{}\fi

6.18.3 Setup

\zref@newlist{savepos}
\zref@newprop*{posx}[0]{\the\pdflastxpos}
\zref@newprop*{posy}[0]{\the\pdflastypos}
\zref@addprops{savepos}{posx,posy}

6.18.4 User macros

\zref@savepos
\def\zref@savepos{\if@filesw\pdfsavepos\fi}
\ZREF@zsavepos
\def\ZREF@zsavepos#1#2#3{\@bsphack\if@filesw\zref@savepos#1{#3}{#2}\ltx@IfUndefined{TeXeTstate}{\ifnum\TeXXeTstate=\ltx@zero\else\zref@savepos\fi}{\fi}\@esphack}

\ZREF@zssavepos
\def\ZREF@zssavepos#1#2#3{\@bsphack\if@filesw\zref@savepos\#1\{#3\}{#2}\#1{#3}\{#2\}\%\ltx@IfUndefined{TeXeTstate}{\ifnum\TeXXeTstate=\ltx@zero\else\zref@savepos\fi}{\else\zref@savepos\fi}\@esphack}

69
The horizontal and vertical position are available by \zposx and \zposy. Do not rely on absolute positions. They differ in DVI and PDF mode of pdfTeX. Use differences instead. The unit of the position numbers is sp.

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

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

\ZREF@savepos@ok

\let\ZREF@savepos@ok=Y
\endinput
\def\zref@def@absposy\ZREF@Robust\def\zref@def@absposy#1{% 
\zref@wrapper@babel{\ZREF@def@abspos{#1}\zref@absposy}%
\}

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

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

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

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

\def\ZREF@abspos@used#1{% 
\zref@refused{#1}%
\zref@ifrefundefined{#1}{% 
\begingroup
\edef\ZREF@temp{\zref@extractdefault{#1}{abspage}{0}}%
\ifnum\ZREF@temp>\ltx@zero
\zref@refused{thepage\ZREF@temp}%
\else
\@PackageError{zref-abspos}{\string\zref@pos@label@used\ltx@space needs property `abspage'\MessageBreak in label `#1'\MessageBreak}@@ehc
\fi
\endgroup
}%
\}

\newcommand*{\zref@absposnumused}{% 
\ifnum#1>\ltx@zero
\zref@refused{thepage\number#1}%
\else
\@PackageError{zref-abspos}{Invalid absolute page number (#1)\MessageBreak A positive integer is expected\MessageBreak}@@ehc
\fi
\}

\usepackage{zref-abspos}
\begin{document}
\zref@absposused{label}
\end{document}
6.19.2 Media

\ZREF@abspos@media@width

\ZREF@abspos@media@height
6.19.3 Paper
6.19.4 Origin

\ZREF@abspos@origin@x

2714 \let\ZREF@temp\ltx@two
2715 \ltx@IfUndefined{pdfhorigin}\{}{%
2716 \ifpdf
2717 \let\ZREF@temp\ltx@zero
2718 \fi
2719 }
2720 \ifx\ZREF@temp\ltx@two
2721 \ifnum\mag=1000\%
2722 \let\ZREF@temp\ltx@one
2723 \fi
2724 \fi
2725 \ifcase\ZREF@temp
2726 \def\ZREF@abspos@origin@x#1{%
2727 \zref@extract{#1}{pdfhorigin}%
2728 }%
2729 \or
2730 \def\ZREF@abspos@origin@x#1{%
2731 4736286%
2732 }%
2733 \or
2734 \def\ZREF@abspos@origin@x#1{%
2735 \numexpr\mag/1000*\dimexpr1truein\relax\relax
2736 }%
2737 \fi

\ZREF@abspos@origin@y

2738 \let\ZREF@temp\ltx@two
2739 \ltx@IfUndefined{pdfvorigin}\{}{%
2740 \ifpdf
2741 \let\ZREF@temp\ltx@zero
2742 \fi
2743 }
2744 \ifx\ZREF@temp\ltx@two
2745 \ifnum\mag=1000\%
2746 \let\ZREF@temp\ltx@one
2747 \fi
2748 \fi
2749 \ifcase\ZREF@temp
2750 \def\ZREF@abspos@origin@y#1{%
2751 \zref@extract{#1}{pdfvorigin}%
2752 }%
2753 \or
2754 \def\ZREF@abspos@origin@y#1{%
2755 4736286%
2756 }%
2757 \or
2758 \def\ZREF@abspos@origin@y#1{%
2759 \numexpr\mag/1000*\dimexpr1truein\relax\relax
2760 }%
2761 \fi

6.19.5 Header

\ZREF@abspos@head@x@left

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

ZREF@abspos@foot@x@left

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

ZREF@abspos@foot@x@right

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

ZREF@abspos@foot@x@center

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

ZREF@abspos@foot@y@bottom

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

6.19.8 Marginal notes

ZREF@abspos@marginpar@x@left

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

ZREF@abspos@marginpar@x@right

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

ZREF@abspos@marginpar@x@center

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

ZREF@abspos@marginpar@y@top

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

ZREF@abspos@marginpar@y@bottom

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

ZREF@abspos@marginpar@y@center

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

77
6.19.9 Stock paper

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

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

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

The definitions of the keys follow.

Defaults are set, see user interface.

Configuration of \zdotfill is done by \zdotfillsetup.

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

\ZREF@IfDefinable{\zdotfill}{\def{\leavevmode}{\global\advance\c@zref@unique\ltx@one\begingroup\def\ZREF@temp{zref@number\c@zref@unique}\pdfsavepos\zref@labelbyprops{\thezref@unique L}{posx}\setlength{\dimen@}{\ZREF@df@unit}\zref@ifrefundefined{\thezref@unique R}{\ZREF@dotfill}{\ifnum\numexpr\zposx{\thezref@unique R}-\zposx{\thezref@unique L}\relax<\dimexpr\ZREF@df@min\dimen@\relax\hfill\ZREF@dotfill}\pdfsavepos\zref@labelbyprops{\thezref@unique R}{posx}\endgroup\kern\z@}}{\ZREF@dotfill}\ZREF@dotfill Help macro that actually sets the dots.

\def{\ZREF@dotfill}{\cleaders{\hb@xt{\dimen@}{\hss\ZREF@df@dot\hss}}\ltx@empty}\langle/\dotfill\rangle

6.21 Module env

\NeedsTeXFormat{LaTeX2e}\ProvidesPackage{zref-env}[2016/05/21 v2.26 Module env for zref (HO)]\RequirePackage{zref-base}[2016/05/21]\ifx\ZREF@base@ok Y\else\expandafter\endinput\fi\zref@newprop{envname}[@currenvir]\zref@newprop{envline}[@currenvline]\ZREF@ENV@line \def{\ZREF@ENV@line}{\expandafter{\ZREF@ENV@line}{\ifx\@currenvline\ltx@empty\else\expandafter{\ZREF@ENV@line}{\ltx@empty}\fi}}\langle/env\rangle

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

\ZREF@ENV@line \def{\ZREF@ENV@line}{1line \ifx\@currenvline\ltx@empty\else\expandafter{\ZREF@ENV@line}{\ltx@empty}\fi\if\@nil\else\expandafter{\ZREF@ENV@line}{\ltx@empty\@nil}\fi}\langle/env\rangle
7 Test

7.1 \zref@localaddprop

\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage{zref-base, qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\makeatletter
\def\ExpectList#1#2{\expandafter\expandafter\expandafter\Expect\expandafter\expandafter\expandafter{\csname Z@L@#1\endcsname}{#2}}
\begin{qstest}{localaddprop}{localaddprop}
\ExpectList{main}{\default\page}\
\Expect{undefined}*{\meaning\foobar}\
\zref@newprop{foobar}{FOO}\
\Expect{undefined}*{\meaning\foobar}\
\% \zref@newlist{alist}\
\ExpectList{alist}{}\
\begingroup
\zref@localaddprop{main}{foobar}\
\Expect{undefined}*{\meaning\foobar}\
\ExpectList{main}{\default\page\foobar}\
\zref@localaddprop{alist}{page}\
\ExpectList{alist}{\page}\
\endgroup
\ExpectList{main}{\default\page}\
\ExpectList{alist}{}\
\zref@addprop{alist}{foobar}\
\ExpectList{alist}{\foobar}\
\Expect{undefined}*{\meaning\foobar}\
\end{qstest}
\@@end

7.2 Module base

\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage{zref-base, zref-titleref}
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\makeatletter
\newcommand*{\DefExpand}[2]{\expandafter\expandafter\expandafter\def\expandafter\expandafter\expandafter#1{#2}\
\@onelevel@sanitize#1}
\newcommand*{\Test}[3]{\Expect{#2}*{#1}\
\zref@wrapper@unexpanded{\Expect*{#3}*{#1}\
\DefExpand\x{#1}}}

\begin{qstest}{localaddprop}{localaddprop}
\DefExpand\x{#1}

80
\section{Hello World}

\label{sec:hello}

\begin{document}

\begin{qstest}{getcurrent}{getcurrent}
Test{\zref@getcurrent{foo}}%
{Value}{\noexpand\@empty V\noexpand\@empty value}%
Test{\zref@getcurrent{xy}}{}{}% 
\end{qstest}

\begin{qstest}{extract}{extract}
\def\textbf#1{<#1>}%
\def\textit#1{[#1]}% hash-ok
Test{\zref@extractdefault{xy}{page}{\@empty D\@empty fault}}%
\Test{\zref@extractdefault{sec:hello}{foo}{\@empty A\@empty B}}%
\Test{\zref@extractdefault{sec:hello}{foo}{\@empty A\@empty B}}%
\Test{\zref@extract{sec:hello}{default}}{1}{1}%
\Test{\zref@extract{sec:hello}{title}}{\[Hello\] <World>}%
\zref@ifrefundefined{sec:hello}{% 
\begin{document}
\begin{qstest}{zruns-document}{zruns-document}
\Expect*{\number\ExpectRuns}*{\zruns}%
\end{qstest}
\end{document}
\end{test}

\section{Module \texttt{runs}}

\begin{test}
\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage{zref-runs}[2016/05/21]
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\begin{qstest}{zruns-preamble}{zruns-preamble}
\Expect{0}*{\zruns}%
\end{qstest}
\AtBeginDocument{% 
\begin{qstest}{zruns-atbegindocument}{zruns-atbegindocument}
\Expect*{\number\ExpectRuns}*{\zruns}%
\end{qstest} 
\begin{document}
\begin{qstest}{zruns-document}{zruns-document}
\Expect*{\number\ExpectRuns}*{\zruns}%
\end{qstest}
\end{document}
\end{test}

\section{Module \texttt{titleref}}

\begin{test}
\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage{zref-titleref-memoir}
\begin{document}
\begin{qstest}{zruns-document}{zruns-document}
\Expect*{\number\ExpectRuns}*{\zruns}%
\end{qstest}
\end{document}
\end{test}
\NeedsTeXFormat{LaTeX2e}
\documentclass{memoir}
\usepackage[zref-titleref][2016/05/21]
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\begin{document}
\def\List{ }
\def\Label{#1}{% 
  \zref@label{#1}%
  \g@addto@macro \List{\par
  \#1: [\ztitleref{#1}]}%}
\mbox{}%
\zref@refused{#1}%
\zref@ifrefundefined{#1}{% 
  \begingroup\edef\x{\zref@extract{#1}{title}}\Expect{OK/}*{}\endgroup}{%}
\endgroup}
\def\Test{#1}{% 
  \csname#1\endcsname*{OK/#1}%
  \Label{#1*}%
  \csname#1\endcsname{OK/#1}%
  \Label{#1}%
  \csname#1\endcsname[OK/#1-toc]%
  \[WRONG-in-titleref/#1-toc-2]%
  \Label{#1-toc}%
  \expandafter\ifx\csname#1\endcsname\part
  \else
  \headnamereffalse
  \csname#1\endcsname[OK/#1-th-toc]%
  \[WRONG-in-titleref/#1-th-toc-2]%
  \Label{#1-th-toc}%
  \headnamereftrue
  \csname#1\endcsname[WRONG-in-titleref/#1-th-head-1]%
  \[OK/#1-th-head]%
  \Label{#1-th-head}%
  \fi
\begin{qstest}{section}{section}
\@for\x:=part,chapter,section,subsection,subsubsection\do{% 
  \expandafter\Test{\expandafter\x}%
}\end{qstest}
\newpage
\List
\end{document}
8 Installation

8.1 Download

Package. This package is available on CTAN:\(^2\):


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

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

TDS refers to the standard “A Directory Structure for \TeX{} Files” (CTAN:tds/tds.pdf). Directories with \texttt{texmf} in their name are usually organized this way.

8.2 Bundle installation

Unpacking. Unpack the \texttt{oberdiek.tds.zip} in the TDS tree (also known as \texttt{texmf} tree) of your choice. Example (linux):

\begin{verbatim}
unzip oberdiek.tds.zip -d ~\texttt{/texmf}
\end{verbatim}

Script installation. Check the directory TDS:scripts/oberdiek/ for scripts that need further installation steps. Package attachfile2 comes with the Perl script \texttt{pdfatfi.pl} that should be installed in such a way that it can be called as \texttt{pdfatfi}.

Example (linux):

\begin{verbatim}
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
\end{verbatim}

8.3 Package installation

Unpacking. The \texttt{.dtx} file is a self-extracting \texttt{docstrip} archive. The files are extracted by running the \texttt{.dtx} through plain \TeX{}:

\begin{verbatim}
tex zref.dtx
\end{verbatim}

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

\(^2\text{http://ctan.org/pkg/zref}\)
If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

### 8.4 Refresh file name databases

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

### 8.5 Some details for the interested

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

- **plain \TeX:** Run `docstrip` and extract the files.
- **\LaTeX:** Generate the documentation.

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

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

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

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

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

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

```
pdf\latex zref.dtx
makeindex -s gind.ist zref.idx
pdf\latex zref.dtx
makeindex -s gind.ist zref.idx
pdf\latex zref.dtx
```

84
9 Catalogue

The following XML file can be used as source for the \TeX Catalogue. The elements \texttt{caption} and \texttt{description} are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is \texttt{zref.xml}.

3090 ⟨*\texttt{catalogue}⟩
3091 <?xml version='1.0' encoding='us-ascii'?>
3092 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
3093 <entry datestamp='$Date$' modifier='$Author$' id='zref'>
3094 <name>zref</name>
3095 <caption>A new reference scheme for \LaTeX.</caption>
3096 <authorref id='auth:oberdiek'/>
3097 <copyright owner='Heiko Oberdiek' year='2006-2012'/>
3098 <license type='lppl1.3'/>
3099 <version number='2.26'/>
3100 <description>
3101 This package offers a means to remove the limitation, of only two
3102 properties, that is inherent in the way \LaTeX's reference system
3103 works. The package implements an extensible referencing system,
3104 where properties are handled in a more flexible way. It provides
3105 an interface for macro programmers to access the new reference
3106 scheme and some applications that use it.
3107 <p/>
3108 The package is part of the \texttt{<xref refid='oberdiek'>oberdiek</xref> bundle.}
3109 <</description>
3110 <documentation details='Package documentation'
3111 href='ctan:/macros/latex/contrib/oberdiek/zref.pdf'/>
3112 <ctan file='true' path='/macros/latex/contrib/oberdiek/zref.dtx'/>
3113 <miktex location='oberdiek'/>
3114 <texlive location='oberdiek'/>
3115 <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
3116 </entry>
3117 ⟨/\texttt{catalogue}⟩

10 References

[1] Package \texttt{footmisc}, Robin Fairbairns, 2004/01/23 v5.3a.\texttt{CTAN:macros/latex/contrib/footmisc/footmisc.dtx}


[3] Package \texttt{lastpage}, Jeff Goldberg, 1994/06/25 v0.1b.\texttt{CTAN:macros/latex/contrib/lastpage/}


[6] Package \texttt{titleref}, Donald Arsenau, 2001/04/05 v3.1.\texttt{CTAN:macros/latex/contrib/misc/titleref.sty}


11 History

[2006/02/20 v1.0]
  • First version.

[2006/05/03 v1.1]
  • Module perpage added.
  • Module redesign as packages.

[2006/05/25 v1.2]
  • Module dotfillmin added.
  • Module base: macros \zref@require@uniqe and \thezref@unique added (used by modules titleref and dotfillmin).

[2006/09/08 v1.3]
  • Typo fixes and English cleanup by Per Starback.

[2007/01/23 v1.4]
  • Typo in macro name fixed in documentation.

[2007/02/18 v1.5]
  • \zref@getcurrent added (suggestion of Igor Akkerman).
  • Module savepos also supports X\TeX{}.

[2007/04/06 v1.6]
  • Fix in modules abspage and base: Now counter abspage and zref@unique are not remembered by \include.
  • Beamer support for module titleref.

[2007/04/17 v1.7]
  • Package atbegshi replaces everyshi.

[2007/04/22 v1.8]
  • \zref@wrapper@babel and \zref@refused are now expandable if babel is not used or \if@safe@actives is already set to true. (Feature request of Josselin Noirel)

[2007/05/02 v1.9]
  • Module titleref: Some support for \caption of package longtable, but only if \label is given after \caption.

[2007/05/06 v2.0]
  • Uses package etexcmds for accessing $\varepsilon$-\TeX’s \unexpanded.
[2007/05/28 v2.1]
- Module titleref supports caption of package listings.
- Fixes in module titleref for support of packages titlesec and longtable.

[2008/09/21 v2.2]
- Module base: \zref@iflistcontainsprop is documented, but a broken \zref@listcontainsprop implemented. Name and implementation fixed (thanks Ohad Kammar).

[2008/10/01 v2.3]
- \zref@localaddprop added (feature request of Ohad Kammar).
- Module lastpage: list ‘LastPage’ added. Label ‘LastPage’ will use the properties of this list (default is empty) along with the properties of the main list.

[2009/08/07 v2.4]
- Module runs added.

[2009/12/06 v2.5]
- Module lastpage: Uses package atveryend.
- Module titleref: Further commands are disabled during string expansion, imported from package nameref.

[2009/12/07 v2.6]
- Version date added for package atveryend.

[2009/12/08 v2.7]
- Module titleref: Use of package gettitlestring.

[2010/03/26 v2.8]
- \zifrefundefined added.
- Module lastpage: Macros \zref@iflastpage and \ziflastpage added.
- Module thepage added.
- Module nextpage added.

[2010/03/29 v2.9]
- Module marks added (without documentation).
- \zref@addprop now adds expanded property to list.
- Useless \ZREF@ErrorNoLine removed.

[2010/04/08 v2.10]
- Module xr remembers the external document name in property ‘externaldocument’.

87
[2010/04/15 v2.11]

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

[2010/04/17 v2.12]

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

[2010/04/19 v2.13]

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

[2010/04/22 v2.14]

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

[2010/04/23 v2.15]

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

[2010/04/28 v2.16]

- More consequent use of package ‘ltxcmds’ and ‘hologo’.
- Module pagelayout added.
- Module pageattr added.
- Robustness introduced for non-expandable interface macros.
- Internal change of the data format of property lists (suggestion of Florent Chervet).
- Module titleref: Support of environment description.
[2010/05/01 v2.17]
- \zref@newprop throws an error if the property already exists.
- Module xr: Bug fix for the case of several .aux files (bug found by Victor Ivrii).
- Module xr: Property ‘urluse’ and option urluse added.

[2010/05/13 v2.18]
- Module env added.
- Module savepos: \zref@savepos added.

[2010/10/22 v2.19]
- \zref@addprop and \zref@localaddprop are limited to one property only (incompatibility to versions v2.13 to v2.18).
- \zref@addprops and \zref@localaddprops added.
- \zref@delprop and \zref@localdelprop added.
- \zref@labelbykv and \zkvlabel (module user) with keys prop, list, delprop, immediate, values added.

[2011/02/12 v2.20]
- Fix for warning in zref-xr.

[2011/03/18 v2.21]
- Fix in module pagelayout for \zlistpagelayout.
- Fix for \zref@localaddprop (probably since v2.19).

[2011/10/05 v2.22]
- Documentation fixed for \zref@(local)addprop(s).
- Module base: \zref@def@extract, \zref@def@extractdefault added.
- Fix in module pagelayout: Because of missing \noexpand commands the values of the pagelayout properties on all pages were the values at package loading.
- Module base: \zref@showprop added.

[2011/12/05 v2.23]
- Module savepos: \zsavenpositions and \zsavenpositiony 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.
update zref-savepos for new luatex
12 Index

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

Symbols

\@end ........................................... 2953
\@PackageError .................. 508, 524, 2604, 2617
\@PackageInfo .................. 1962
\@PackageInfoNoLine........ 548, 563, 1258, 1346, 1440, 1453, 1465, 1521, 1566
\@PackageWarning .............. 691
\@addtoreset .......................... 911, 1008
\@auxout ................................ 715
\@begintheorem .................... 2001, 2006
\@bsphack .......................... 600, 610, 630, 2476
\@caption .......................... 1846
\@chapter .......................... 1858, 1895
\@currentHref .......................... 940
\@currentlabel .................... 935
\@currenvir .......................... 2908
\@currenvline .................... 2911, 2914
\@e@hc ................................ 296, 306, 491, 514, 526, 1596, 2608, 2621
\@empty .......................... 2984, 2987, 2993, 2994, 2995, 2996, 2998
\@esphack .................. 607, 627, 642, 2488
\@for .................................. 3082
\@ifclassloaded .................. 1893, 1930
\@ifdefinable .................... 242, 289
\@ifnextchar .......................... 530, 1712
\@ifstar .................................. 495, 2082
\@ifundefined .................... 192, 909, 1723, 2218, 2260
\@input .................................. 2279
\@inputcheck 2110, 2111, 2126, 2163, 2165
\@latex@warning .............. 777
\@mainaux .......................... 1681
\@namedef .................................. 535
\@newl@obei .......................... 285
\@nil .................................. 1611, 1613, 1811, 1818, 2000, 2001, 2154, 2157, 2337, 2346, 2914, 2917, 2918, 2955
\@onelevel@sanitize ............... 422, 440, 505, 533, 1999, 2002, 2968
\@opargbegintheorem ............ 1991
\@opargbegintheorem ............ 1852
\@opargchapter .............. 1876
\@opsect .................................. 1864
\@os@part .......................... 1870
\@ossect ................................ 1882
\@op@testdef .................... 1928, 1929, 1500, 1501
\@op@testopt .................... 2084, 2087, 2097
\@os@for ................................ 322, 720
\@undefined .................... 1767, 2336, 2439
\| ..................................... 25, 26, 27, 28, 153, 155, 157, 158, 170, 173, 2245, 2339, 2364, 2374, 2378, 2394
\u ..................................... 44, 45

\AddLineBeginAux .................. 280
\adv ance .................................. 1053, 1393, 1548, 1736, 2058, 2872
\afterassignment .................. 233, 1137, 1141
\@AfterLastShipout .............. 1050, 1410, 1573
\@Alph .................................. 7
\@anchor .................................. 2376
\@AtBeginDocument .................. 1029, 1226, 1675, 1844, 3021
\@AtBeginShipout .................. 1012, 1097
\@AtBeginShipoutAddToBox ........ 1098
\@AtBeginShipoutBoxDepth .......... 1318
\@AtBeginShipoutBoxHeight ........ 1317
\@AtBeginShipoutBoxWidth .......... 1316
\@AtEndOfPackage .................. 195
\@AtVeryEndDocument .............. 1337, 1509

\beam er@section ..................... 1932
\beam er@subsection .............. 1938
\beam er@subsubsection ............ 1944
\begin .................. 23, 57, 100, 106, 156, 172, 2933, 2980, 2985, 2990, 3017, 3022, 3027, 3028, 3040, 3081
\@bfseries .................................. 928

\@cabspace ...................... 1015, 1744, 1751
\@c@page ...................... 954, 1053, 1741, 2766, 2817
\@c@zpage .......................... 1741, 1747
\@c@zrefer@unique .................. 915, 1736, 2872, 2874
\@ch@pt@C .................. 1899, 1905, 1911
\@chapter .................................. 24, 30, 32, 61, 82
\@ChapterPages .................. 91, 112
\@ChapterStart .................. 78, 135, 150, 166
\@ChapterStop .................. 85, 148, 165, 184
\@ch@rdef ...................... 1159, 1174, 1183, 1187
\@cleaders .......................... 2896
\@cleardoublepage .............. 79, 86
\@clearpage .......................... 62
\@closein .................................. 2126
\@columnsep .......................... 1309
\@columnwidth .......................... 1308
\@comma@entry .................. 343, 344, 346, 352, 378, 379, 381, 387, 614, 616, 620, 1617, 1618, 1619, 1625, 1628
\@comma@parse .................. 342, 377, 613, 1616
\@count@ .................. 1343, 1354, 1355, 1357, 1392, 1393, 1402, 1404, 1405, 1518, 1539, 1541, 1542
\@count@ .................. 1547, 1548, 2052, 2057, 2068, 2069
\@cs@name .................. 252, 253, 290, 315, 316, 317, 326, 351, 352, 369, 370, 386, 387, 404, 405, 425, 427, 444, 462, 465, 478, 536, 538, 539,