Abstract

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

Contents

1 Introduction 4
  1.1 Standard \LaTeX{} behaviour 4
  1.2 Basic idea 5
  1.3 Interfaces 5

2 Interface for programmers 5
  2.1 Entities 5
  2.2 Property list 6
  2.3 Property 7
  2.4 Reference generation 7
  2.5 Data extraction 8
  2.6 Setup 9
  2.7 Declared properties 10
  2.8 Wrapper for advanced situations 11
  2.9 Counter for unique names 11

3 User interface 11
  3.1 Module user 11
  3.2 Module abspage 12
  3.3 Module lastpage 13
    3.3.1 Tests for last page 13
    3.3.2 Example 13
  3.4 Module thepage 14
  3.5 Module nextpage 15
    3.5.1 Configuration 15
    3.5.2 Example 15
  3.6 Module totpages 16
  3.7 Module pagelayout 16
  3.8 Module marks 17
  3.9 Module runs 17
  3.10 Module perpage 17

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

\begin{verbatim}
\xref_{⟨your key⟩}_key{⟨some text⟩}
\end{verbatim}

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 \LaTeX{}3 that will need
some time before its first release. Thus I have implemented it as \LaTeX{}2ε package
without disturbing the existing \LaTeX{} reference system.

1.3 Interfaces

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

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

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

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

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

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

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

\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:

\Z@R@myref → \default{2.1}\page{7}

The generic format of an entry is:

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

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

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

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

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

2.2 Property list

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

\zref@newlist {⟨listname⟩}

Declares a new empty property list.

\zref@addprop {⟨listname⟩} {⟨propname⟩}
\zref@localaddprop {⟨listname⟩} {⟨propname⟩}

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

\zref@listexists {⟨listname⟩} {⟨then⟩}

Executes ⟨then⟩ if the property list ⟨listname⟩ exists or raise an error otherwise.
\zref@iflistundefined\{\\langle\textit{listname}\rangle\} \{\langle\textit{then}\rangle\ \{\langle\textit{else}\rangle\}

Executes \langle\textit{then}\rangle\ if the list exists or \langle\textit{else}\rangle\ otherwise.

\zref@iflistcontainsprop\{\\langle\textit{listname}\rangle\} \{\langle\textit{propname}\rangle\} \{\langle\textit{then}\rangle\ \{\langle\textit{else}\rangle\}

Executes \langle\textit{then}\rangle\ if the property \langle\textit{propname}\rangle\ is part of property list \langle\textit{listname}\rangle\ or otherwise it runs the \langle\textit{else}\rangle\ part.

2.3 Property

\zref@newprop\*\{\langle\textit{propname}\rangle\} \{\langle\textit{default}\rangle\} \{\langle\textit{value}\rangle\}

This command declares and configures a new property with name \langle\textit{propname}\rangle. In case of unknown references or the property does not exist in the reference, the \langle\textit{default}\rangle\ 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\{\langle\textit{propname}\rangle\} \{\langle\textit{value}\rangle\}

This sets the current value of the property \langle\textit{propname}\rangle. It is a generalization of setting \LaTeX’s \texttt{\currentlabel}.

\zref@getcurrent\exp2\{\langle\textit{propname}\rangle\}

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

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

\zref@propexists\{\langle\textit{propname}\rangle\} \{\langle\textit{then}\rangle\}

Calls \langle\textit{then}\rangle\ if the property \langle\textit{propname}\rangle\ is available or generates an error message otherwise.

\zref@ifpropundefined\exp\{\langle\textit{propname}\rangle\} \{\langle\textit{then}\rangle\ \{\langle\textit{else}\rangle\}

Calls \langle\textit{then}\rangle\ or \langle\textit{else}\rangle\ depending on the existence of property \langle\textit{propname}\rangle.

2.4 Reference generation

\zref@label\{\langle\textit{refname}\rangle\}

This works similar to \texttt{\label}. The reference \langle\textit{refname}\rangle\ is created and put into the \texttt{.aux} file with the properties of the main property list.
\zref@labelbylist {⟨refname⟩} {⟨listname⟩}

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

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

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

\zref@newlabel {⟨refname⟩} {...}

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

2.5 Data extraction

\zref@extractdefault\exp2 {⟨refname⟩} {⟨propname⟩} {⟨default⟩}

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

\zref@extract\exp2 {⟨refname⟩} {⟨propname⟩}

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

Example for page references:

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

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

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

\zref@refused {⟨refname⟩}

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

\zref@def@extract {⟨cmd⟩} {⟨refname⟩} {⟨propname⟩}
\zref@def@extractdefault {⟨cmd⟩} {⟨refname⟩} {⟨propname⟩} {⟨default⟩}

Both macros extract the property ⟨propname⟩ from the reference ⟨refname⟩ the same way as macros \zref@extract and \zref@extractdefault. The result is stored in macro ⟨cmd⟩. Also \zref@refused is called to notify \LaTeX that the reference ⟨refname⟩ is used. Added in 2011/10/04 v2.22.
Macro \zref@ifrefundefined calls arguments \{then\} or \{else\} dependent on the existence of the reference \{refname\}.

Macro \zifrefundefined calls \ref@refused before executing \zref@ifrefundefined. Babel shorthands are supported in \{refname\}.

Test whether a reference provides a property.

### 2.6 Setup

\zref@default

Holds the global default for unknown values.

\zref@setdefault \{value\}

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

\zref@setmainlist \{value\}

Sets the name of the main property list. The package sets and uses \texttt{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>(base)</td>
<td>default</td>
<td>main</td>
<td><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td></td>
<td>page</td>
<td>main</td>
<td><code>&lt;empty&gt;</code></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><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td>hyperref</td>
<td>anchor</td>
<td>main</td>
<td><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td></td>
<td>url</td>
<td></td>
<td><code>&lt;empty&gt;</code></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>pagelayout</td>
<td>mag</td>
<td>thepage</td>
<td><code>\number\mag</code></td>
</tr>
<tr>
<td></td>
<td>paperwidth</td>
<td>thepage</td>
<td><code>\number\paperwidth</code></td>
</tr>
<tr>
<td></td>
<td>paperheight</td>
<td>thepage</td>
<td><code>\number\paperheight</code></td>
</tr>
<tr>
<td></td>
<td>stockwidth</td>
<td>thepage</td>
<td><code>\number\stockwidth</code></td>
</tr>
<tr>
<td></td>
<td>stockheight</td>
<td>thepage</td>
<td><code>\number\stockheight</code></td>
</tr>
<tr>
<td></td>
<td>pdfpagewidth</td>
<td>thepage</td>
<td><code>\number\pdfpagewidth</code></td>
</tr>
<tr>
<td></td>
<td>pdfpageheight</td>
<td>thepage</td>
<td><code>\number\pdfpageheight</code></td>
</tr>
<tr>
<td></td>
<td>pdfhorigin</td>
<td>thepage</td>
<td><code>\number\pdfhorigin</code></td>
</tr>
<tr>
<td></td>
<td>pdfvorigin</td>
<td>thepage</td>
<td><code>\number\pdfvorigin</code></td>
</tr>
<tr>
<td></td>
<td>hoffset</td>
<td>thepage</td>
<td><code>\number\hoffset</code></td>
</tr>
<tr>
<td></td>
<td>voffset</td>
<td>thepage</td>
<td><code>\number\voffset</code></td>
</tr>
<tr>
<td></td>
<td>topmargin</td>
<td>thepage</td>
<td><code>\number\topmargin</code></td>
</tr>
<tr>
<td></td>
<td>oddside margin</td>
<td>thepage</td>
<td><code>\number\oddsidemargin</code></td>
</tr>
<tr>
<td></td>
<td>evensidemargin</td>
<td>thepage</td>
<td><code>\number\evensidemargin</code></td>
</tr>
<tr>
<td></td>
<td>textwidth</td>
<td>thepage</td>
<td><code>\number\textwidth</code></td>
</tr>
<tr>
<td></td>
<td>textheight</td>
<td>thepage</td>
<td><code>\number\textheight</code></td>
</tr>
<tr>
<td></td>
<td>headheight</td>
<td>thepage</td>
<td><code>\number\headheight</code></td>
</tr>
<tr>
<td></td>
<td>headsep</td>
<td>thepage</td>
<td><code>\number\headsep</code></td>
</tr>
<tr>
<td></td>
<td>footskip</td>
<td>thepage</td>
<td><code>\number\footskip</code></td>
</tr>
<tr>
<td></td>
<td>marginparwidth</td>
<td>thepage</td>
<td><code>\number\marginparwidth</code></td>
</tr>
<tr>
<td></td>
<td>marginparsep</td>
<td>thepage</td>
<td><code>\number\marginparsep</code></td>
</tr>
<tr>
<td></td>
<td>columnwidth</td>
<td>thepage</td>
<td><code>\number\columnwidth</code></td>
</tr>
<tr>
<td></td>
<td>columns sep</td>
<td>thepage</td>
<td><code>\number\columnsep</code></td>
</tr>
<tr>
<td>perpage</td>
<td>page value</td>
<td>perpage</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>page</td>
<td>perpage</td>
<td><code>&lt;empty&gt;</code></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><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td>xref</td>
<td>anchor</td>
<td></td>
<td><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td></td>
<td>external document</td>
<td></td>
<td><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td></td>
<td>theotype</td>
<td></td>
<td><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td></td>
<td>title</td>
<td></td>
<td><code>&lt;empty&gt;</code></td>
</tr>
<tr>
<td></td>
<td>url</td>
<td></td>
<td><code>&lt;empty&gt;</code></td>
</tr>
</tbody>
</table>

1 Module `pagelayout` only defines properties if the parameter exists.
2.8 Wrapper for advanced situations

\zref@wrapper@babel {...} \{\{name\}\}

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

\zref@wrapper@immediate {...}

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

\zref@wrapper@unexpanded {...}

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

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

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

2.9 Counter for unique names

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

\zref@require@unique

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

\thezref@unique

This command is used to generate unique label names.

3 User interface

3.1 Module user

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

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

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

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

\begin{verbatim}
\zlabel{⟨refname⟩}{blabel}
Similar to \texttt{\label}. It generates a label with name \texttt{⟨refname⟩} in the new reference scheme.
\end{verbatim}

\begin{verbatim}
\zref{⟨propname⟩}{⟨refname⟩}{blabel}
Without optional argument similar to \texttt{\ref}, it returns the default reference property. This property is named \texttt{default}:
\begin{verbatim}
\zref{x} ≡ \zref[default]{x}
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\zpageref{⟨refname⟩}{blabel}
Convenience macro, similar to \texttt{\pageref}.
\begin{verbatim}
\zpageref{x} ≡ \zref[page]{x}
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\zrefused{⟨refname⟩}{blabel}
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{\refused} is strongly recommended. The reference \texttt{⟨refname⟩} is marked as used, undefined ones will generate warnings.
\end{verbatim}

3.2 Module abspage

With the help of package \texttt{atbegshi} a new counter \texttt{abspage} with absolute page numbers is provided. For technical and historical reasons the counter itself is zero based: if you use it directly on the first page, e.g with \texttt{\arabic{abspage}} you will get 0 as value. When using \texttt{\zref} the first page will be page 1 as expected. Also a new property \texttt{abspage} is defined and added to the main property list. Thus you can reference the absolute page number:

\begin{verbatim}
Section \zref{foo} is on page \zpageref{foo}.
This is page \zref[abspage]{foo}
of \zref[abspage]{LastPage}.
\end{verbatim}
The example also makes use of module `lastpage`.

### 3.3 Module `lastpage`

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

\[ \zref@extract{LastPage}{\text{page}} (+ \zref@refused{LastPage}) \]

or

\[ \zpageref{LastPage} \text{(module user)} \]

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.

\[
\zref@iflastpage\{\langle refname\rangle\}\{\langle then\rangle\}\{\langle else\rangle\}
\]

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

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

\[
\ziflastpage\{\langle refname\rangle\}\{\langle then\rangle\}\{\langle else\rangle\}
\]

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

#### 3.3.2 Example

```latex
\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}
\documentclass{report}
\newcounter{foo}
\renewcommand*{\thefoo}{\Alph{foo}}
\usepackage{zref-lastpage,zref-user}\[2019/11/29\]
\makeatletter
\end{verbatim}
```
3.4 Module thepage

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

\zthepage\{(absolute page number)\}

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

\zref@thepage@name\{exp\}\{(absolute page number)\}

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

\zref@thepage\{exp\}\{(absolute page number)\}
\zref@thepage@refused\{(absolute page number)\}

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

\znextpage

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

1. The next page is not known yet because of undefined references. Then \zunknownnextpagename is used instead. The default for this macro is the default of property page.

2. This page is the last page. Then \znonextpagename is used. Its default is empty.

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

3.5.1 Configuration

The behaviour can be configured by the following macros.

\zunknownnextpagename
\znonextpagename

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

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

\znextpagesetup \{\langle\text{unknown}\rangle\} \{\langle\text{no next}\rangle\} \{\langle\text{next}\rangle\}

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

\{\text{*example-nextpage}\}
\%<<END_EXAMPLE
\documentclass{book}

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

15
3.6 Module `totpages`

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

```
\ztotpages^{exp}
```

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

3.7 Module `pagelayout`

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

Some parameters are specific for a class (e.g. `stockwidth` and `stockheight` for class `memoir`) or the \TeX engine like 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 `thepage` is loaded that generates a label for each page. The properties of module `pagelayout` are added to the property list `thepage` of module `thepage`.

List of properties:
\zlistpagelayout

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

3.8 Module marks

ToDo.

3.9 Module runs

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

\zruns\exp

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

3.10 Module perpage

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

\begin{tabular}{lll}
\textbf{Parameter} & \textbf{Property} & \textbf{Remarks} \\
\mag & mag & \\
\linewidth & \linewidth & \\
\paperwidth & paperwidth & \\
\paperheight & paperheight & \\
\stockwidth & stockwidth & class memoir \\
\stockheight & stockheight & class memoir \\
\pdfpagewidth & pdfpagewidth & \pdfTeX, \luaTeX{} \\
\pdfpageheight & pdfpageheight & \pdfTeX, \luaTeX{} \\
\pdfhorigin & pdfhorigin & \pdfTeX, \luaTeX{} \\
\pdfvorigin & pdfvorigin & \pdfTeX, \luaTeX{} \\
\hoffset & hoffset & \\
\voffset & voffset & \\
\topmargin & topmargin & \\
\oddsidemargin & oddsidemargin & \\
\evensidemargin & evensidemargin & \\
\textwidth & textwidth & \\
\textheight & textheight & \\
\headheight & headheight & \\
\headsep & headsep & \\
\footskip & foottip & \\
\marginparwidth & marginparwidth & \\
\marginparsep & marginparsep & \\
\columnwidth & columnwidth & \\
\columnsep & columnsep &
\end{tabular}
somewhere on the next page. A reference mechanism costs at least two \LaTeX runs, but ensures correct page counter values.

\makeperpage {\reset} {\counter}

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

\makeperpage{footnote} % \usepackage[perpage]{footmisc}

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

\makeperpage[2]{footnote}

\thezpage

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

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

\unmakeperpage {\counter}

The reset mechanism for this counter is deactivated.

3.11 Module counter

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

3.12 Module titleref

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

\titleref {\refname}

Print the section or caption title of reference \refname, similar to \nameref or \titleref.
This command allows to configure the behaviour of module \texttt{titleref}. The following keys are available:

\begin{verbatim}
\zetitlerefsetup\{key_1=value_1, key_2=value_2, \ldots\}
\end{verbatim}

\texttt{title}=(value)

Sets the current title.

\texttt{stripperiod=true|false}

Follow package \texttt{nameref} that removes a last period. Default: \texttt{true}.

\texttt{expand=true|false}

Package \texttt{titleref} expands the title first. This way garbage and dangerous commands can be removed, e.g. \texttt{\label}, \texttt{\index}… See implementation section for more details. Default is \texttt{false}.

\texttt{cleanup=\ldots}

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

\subsection{Module savepos}

This option supports a feature that \texttt{pdfT\TeX} provides (and X\texttt{T\TeX}). \texttt{pdfT\TeX} is able to tell the current position on the page. The page position is not instantly known. First the page must be constructed by \texttt{T\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.

\begin{verbatim}
\zsavepos\{⟨refname⟩\}
\end{verbatim}

It generates a reference with name \langle\texttt{refname}\rangle. The reference stores the location where \zsavepos is executed in properties \texttt{posx} and \texttt{posy}.

\begin{verbatim}
\zsavepos\{⟨refname⟩\}
\zsavepos\{⟨refname⟩\}
\zsavepos\{⟨refname⟩\}
\end{verbatim}

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

\begin{verbatim}
\zposx\exp\{⟨refname⟩\}
\zposy\exp\{⟨refname⟩\}
\end{verbatim}

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

Do not rely on absolute page numbers. Because of problems with the origin the numbers may differ in DVI or PDF mode of \texttt{pdfT\TeX}. Therefore work with relative values by comparisons.

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

This module uses \texttt{pdfT\TeX}’s \texttt{pdfsavepos}, \texttt{pdflastxpos}, and \texttt{pdflastypos}. They are available in PDF mode and since version 1.40.0 also in DVI mode.
Macro \zref{savepos} performs the first part of \zsavepos by calling \pdfsavepos (if .aux files are writable).

Thus \zsavepos is basically \zref{savepos} followed by \zref{labelbylist}{⟨refname⟩}{savepos}. If \TeXstate is detected and enabled, \savepos also adds \zref{savepos} at the end to support \begin{R} where the whatits are processed in reverse order. The property list \savepos contains the properties \texttt{posx} and \texttt{posy}.

3.14 Module \texttt{abspos}

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

\zref{absposx}{⟨label⟩}{⟨value⟩}{⟨position⟩}
\zref{absposy}{⟨label⟩}{⟨value⟩}{⟨position⟩}

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

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

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

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

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

3.15 Module \texttt{dotfill}

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

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

This command allows to configure the behaviour of \zdotfill. The following keys are available:
\min = \langle \text{count value} \rangle

If the actual number of dots are smaller than \langle \text{count value} \rangle, then the dots are suppressed. Default: 2.

\unit = \langle \text{dimen value} \rangle

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

\dot = \langle \text{value} \rangle

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

\subsection{Module 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.

\subsection{Module xr}

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

\begin{verbatim}
\externaldocument \[ \langle \text{prefix} \rangle \] \{\langle \text{external document} \rangle\} \[ \langle \text{url} \rangle \]
\end{verbatim}

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

In the star form it tries to detect labels from \texttt{hyperref}, \texttt{titleref}, and \texttt{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 \texttt{anchor} properties, then we need also a \texttt{url} to be able to address the external file. As default the filename is taken with a default extension.

\begin{verbatim}
\zxrsetup {key_1=value_1, key_2=value_2, \ldots}
\end{verbatim}

The following setup options are available:

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

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

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

\texttt{urluse}: Boolean option. If enabled, then a URL is stored in a macro and the macro is put in property \texttt{urluse}. The URL is not put in property \texttt{url}. The purpose is to save \TeX memory.
verbose: Boolean option. List the imported labels in the .log file. Default is false.

\zref@xr@ext

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

3.18 Module pageattr

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

\zref@pdfpageattr{⟨absolute page number⟩}
\zref@pdfpagesattr{⟨absolute page number⟩}

4 ToDo

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

- Other applications: autoref, hyperref, ...

5 Example

67 (*example)
68 \documentclass{book}
69 \usepackage[ngerman]{babel}\
70 \usepackage[savepos,totpages,titleref,dotfill,counter,user]{zref}
71
72 Chapters are wrapped inside \ChapterStart and \ChapterStop. The first argument #1 of \ChapterStart is used to form a label id chap:#1. At the end of the chapter another label is set by \zref@wrapper@immediate, because otherwise at the end of document a deferred write would not be written, because there is no page for shipout.

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

74 \makeatletter
75 \zref@newprop{chaptitle}{}
76 \zref@addprop{main}{chaptitle}
77
78 \newcommand*{\ChapterStart}[2]{%\
79 \cleardoublepage\
80 \def\current@chapid{#1}\
81 \zref@setcurrent{chaptitle}{#2}\
82 \chapter[#2]\
83 \zlabel{chap:#1}\
84 }\
85 \newcommand*{\ChapterStop}{%
\chapter{ChapterPages} calculates and returns the number of pages of the referenced chapter.

\newcommand*{\ChapterPages}[1]{\zref{chap:#1}\zref{chapend:#1}\number\numexpr\zref{chapend:#1}{abspage}\ -\zref{chap:#1}{abspage}\ +1\relax}

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

The frontmatter part has \number\numexpr\zref{chap:first}{abspage}\ -1\relax \textemdash\ pages.

Chapter \zref{chap:first} has \ChapterPages{first} page(s).

Section \zref{hello} is on the \ifcase\numexpr\zref{extractdefault{hello}{page}}\ -\zref{extractdefault{chap:first}{page}}\ +1\relax \or first\or second\or third\or forth\fi page inside its chapter.

The document has \zref[abspage]{LastPage} pages. This number is \ifodd\ztotpages odd\else even\fi.

The last page is labeled with \zpageref{LastPage}.

The title of chapter \zref{chap:next} is ‘‘\zref[chaptitle]{chap:next}’’.

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}.
\newpage
\section{Hello World}
\zlabel{hello}
\ChapterStop
\ChapterStart{next}{Next chapter with \emph{umlauts}: "a"o"u"s}

Here an example follows that makes use of pdf\LaTeX’s “savepos” feature. The position on the page is not known before the page is constructed and shipped out. Therefore the position is stored in references and are available for calculations in the next \LaTeX compile run.
\begin{tabular}{ll}
\zsavepos{firstcol}Hello&\zsavepos{secondcol}World\\
\zsavepos{secondline}Second line&foobar\\
\end{tabular}

With \zrefused \LaTeX is notified, if the references are not yet available and \LaTeX can generate the rerun hint.
\zrefused{firstcol}
\zrefused{secondcol}
\zrefused{secondline}
\ChapterStop

Test for module \dotfill.
\begin{tabular}{rr}
& \verb|\dotfill| & \verb|\zdotfill|\\
\dftest{0.43em} & & \\
\dftest{0.44em} & & \\
\dftest{0.45em} & & \\
\dftest{0.87em} & & \\
\dftest{0.88em} & & \\
\dftest{0.89em} & & \\
\dftest{1.31em} & & \\
\dftest{1.32em} & & \\
\dftest{1.33em} & & \\
\end{tabular}
\ChapterStop
6 Implementation

6.1 Package zref

6.1.1 Identification

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref} [2020-05-28 v2.31 A new reference scheme for LaTeX (HO)]

6.1.2 Load basic module

\RequirePackage{zref-base}[2019/11/29]
Abort package loading if zref-base could not be loaded successfully.
\@ifundefined{ZREF@base@ok}{\endinput}{}

6.1.3 Process options

Known modules are loaded and the release date is checked.
\def\ZREF@temp#1{%
  \DeclareOption{#1}{%
    \AtEndOfPackage{%
      \RequirePackage{zref-#1}[2019/11/29]%
    }%
  }%
} %
\ZREF@temp{abspage}
\ZREF@temp{counter}
\ZREF@temp{dotfill}
\ZREF@temp{hyperref}
\ZREF@temp{lastpage}
\ZREF@temp{marks}
\ZREF@temp{nextpage}
\ZREF@temp{pageattr}
\ZREF@temp{pagelayout}
\ZREF@temp{perpage}
\ZREF@temp{runs}
\ZREF@temp{savepos}
\ZREF@temp{thepage}
\ZREF@temp{titleref}
\ZREF@temp{totpages}
\ZREF@temp{user}
\ZREF@temp{xr}
\ProcessOptions\relax
(/package)

6.2 Module base

6.2.1 Prefixes

This package uses the following prefixes for macro names:
\zref@: Macros of the programmer’s interface.
\ZREF@: Internal macros.
\Z@L@listname: The properties of the list ⟨listname⟩.
\Z@D@propname: The default value for property \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} % 2020-05-28 v2.31 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

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{\afterassignment\ZREF@temp #1#2%
\ZREF@IfDefinable
\def\ZREF@IfDefinable#1#2#3{% @ifdefinable{#1}{\ZREF@Robust{#2}#1#3} %
\ZREF@Robust#1{%
\protected#1%}
%}
%
\ZREF@UpdatePdfTeX is used as help message text in error messages.
\ZREF@UpdatePdfTeX
246 \def\ZREF@UpdatePdfTeX{Update pdfTeX.}
\ZREF@UpdatePdfTeX
\ifZREF@found
The following switch is used in list processing.
\ZREF@UpdatePdfTeX
\newif\ifZREF@found
\ZREF@UpdatePdfTeX
Macro \ZREF@patch first checks the existence of the command and saves it.
\ZREF@UpdatePdfTeX
246 \def\ZREF@patch#1{%
\let\ltx@ifundefined{#1}{%
\expandafter\gobble
\ltx@letcsname ZREF@org@#1\expandafter\endcsname
\csname #1\endcsname
\expandafter\endinput
}
}
6.2.4 Check for \TeX
The use of \TeX should be standard nowadays for \LaTeX. We test for \TeX in order to use its features later.
\ltx@ifundefined{eTeXversion}{%
\PackageError{\ZREF@name}{Missing support for \TeX; package is abandoned}%
\expandafter\endinput
}
\RequirePackage{etexcmds}[2007/09/09]
\ifetex@unexpanded
\else
\PackageError{\ZREF@name}{Missing \TeX’s \unexpanded.\MessageBreak
Add \string\RequirePackage\string{etexcmds\string} before \%
\string\documentclass\}
\expandafter\endinput
\fi
6.2.5 Auxiliary file stuff
We are using some commands in the .aux files. However sometimes these auxiliary files are interpreted by \LaTeX processes that haven’t loaded this package (e.g. package xr). Therefore we provide dummy definitions.
\RequirePackage{auxhook}
\AddLineBeginAux{%
\string\providecommand\string\zref@newlabel[2]{%}
\}
\ZREF@RefPrefix
\def\ZREF@RefPrefix{Z@R}

27
\texttt{\textbackslash{newlabel}}\footnote{For the implementation of \texttt{\textbackslash{newlabel}} we call the same internal macro \texttt{\textbackslash{newl@bel}} that is used in \texttt{\textbackslash{newlabel}}. Thus we have for free:}

- \texttt{\textbackslash{Z@R@labelname}} is defined.
- \LaTeX{}'s check for multiple references.
- \LaTeX{}'s check for changed references.

\texttt{\textbackslash{ZREF@Robust}}\texttt{\edef{\textbackslash{newlabel}}{}}\texttt{\noexpand\textbackslash{newl@bel}{\ZREF@RefPrefix}}\%

\subsection{Property lists}

\texttt{\textbackslash{newlist}}\footnote{Property lists are stored as list of property names enclosed in curly braces. \texttt{\textbackslash{newlist}} creates a new list as empty list. Assignments to property lists are global.}

\begin{verbatim}
\ZREF@Robust\def{\textbackslash{newlist}}#1{%
  \iflistundefined{#1}{%\PackageInfo{\ZREF@name}{New property list: #1}}{%%
    \PackageError{\ZREF@name}{Property list '#1' already exists}\@ehc}
}\end{verbatim}

\texttt{\textbackslash{iflistundefined}}\footnote{\texttt{\textbackslash{iflistundefined}} checks the existence of the property list \texttt{#1}. If the property list is present, then \texttt{#2} is executed and \texttt{#3} otherwise.}

\begin{verbatim}
\ZREF@Robust\def{\textbackslash{iflistundefined}}#1{%
  \iflistundefined{#1}{%\ltx@ifundefined{Z@L@#1}}{%%
    \PackageError{\ZREF@name}{Property list '#1' does not exist}\@ehc}
}\end{verbatim}

\texttt{\textbackslash{listexists}}\footnote{\texttt{\textbackslash{listexists}} only executes \texttt{#2} if the property list \texttt{#1} exists and raises an error message otherwise.}

\begin{verbatim}
\ZREF@Robust\def{\textbackslash{listexists}}#1#2{%
  \iflistundefined{#1}{%\ltx@ifundefined{Z@L@#1}}{%%
    \PackageError{\ZREF@name}{Property list \texttt{#1} does not exist}\@ehc}
}\end{verbatim}

\texttt{\textbackslash{iflistcontainsprop}}\footnote{\texttt{\textbackslash{iflistcontainsprop}} checks, whether a property \texttt{#2} is already present in a property list \texttt{#1}.}

\begin{verbatim}
\ZREF@Robust\def{\textbackslash{iflistcontainsprop}}#1#2{%\ltx@secondoftwo
  \begingroup\expandafter\endgroup
  \expandafter\in@
    \csname#2\expandafter\expandafter\expandafter\endcsname
    \expandafter\expandafter\expandafter{\csname Z@L@#1\endcsname}\csname ltx@ifin@ first\else second\fi oftwo\endcsname
}\end{verbatim}
\zref@listforloop
\def\zref@listforloop#1#2{%
\zref@listexists{#1}{%
\expandafter\expandafter\expandafter\@tfor
\expandafter\expandafter\expandafter\zref@prop
\expandafter\expandafter\expandafter:=%
\csname Z@L@#1\endcsname
\do{%
\begingroup
\escapechar=-1 %
\edef\x{\endgroup
\def\noexpand\zref@prop{%
\expandafter\string\zref@prop
}\x
#2\zref@prop
}%
}%
}%
}%
\zref@addprops
\zref@addprop adds the properties \texttt{#2} to the property list \texttt{#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
\g@addto@macro\csname Z@L@#1\expandafter\endcsname\expandafter{\csname\comma@entry\endcsname}%
}
}\ltx@gobble
}%
\zref@addprop
\zref@addprop adds the property \texttt{#2} to the property list \texttt{#1}, if the property is not already in the list. Otherwise a warning is given.
\ZREF@Robust\def\zref@addprop#1#2{%
\zref@propexists{#2}{%
\zref@iflistcontainsprop{#1}{#2}{%
\PackageWarning\ZREF@name{Property ‘#2’ is already in list ‘#1’}%
}%
}%
}\begingroup\expandafter\endgroup
}
\zref@localaddprops
\ZREF@Robust\def\zref@localaddprops#1#2{%  \zref@listexists{#1} {%  \comma@parse{#2} {%  \zref@propexists{\comma@entry} {{ \zref@iflistcontainsprop{#1} {\comma@entry} { PackageWarning\ZREF@name{ Property \"\comma@entry\" is already in list \"#1\" } } { \begingroup\expandafter\endgroup\expandafter\ltx@LocalAppendToMacro\csname Z@L@#1\expandafter\endcsname\expandafter{\csname\comma@entry\endcsname} } } { \ltx@gobble } } %}
} %
\zref@localdelprop
\ZREF@Robust\def\zref@localdelprop#1#2{%  \zref@listexists{#1} {%  \zref@propexists{#2} {%  \zref@iflistcontainsprop{#1}{#2} { PackageWarning\ZREF@name{ Property \#2 is already in list \#1 } } { \begingroup\expandafter\endgroup\expandafter\ltx@LocalAppendToMacro\csname Z@L@#1\expandafter\endcsname\expandafter{\csname\comma@entry\endcsname}\expandafter{\csname#2\endcsname} } { \ltx@gobble } } %}
} %
\ltx@IfUndefined{pdf@strcmp}{%  \zref@delprop
\ZREF@Robust\def\zref@delprop{%  \ZREF@delprop\gdef %}
} %
\zref@localdelprop
\ZREF@Robust\def\zref@localdelprop#1#2{%  \zref@listexists{#1} {%  \zref@propexists{#2} {%  \zref@iflistcontainsprop{#1}{#2} { PackageWarning\ZREF@name{ Property \#2 is already in list \#1 } } { \begingroup\expandafter\endgroup\expandafter\ltx@LocalAppendToMacro\csname Z@L@#1\expandafter\endcsname\expandafter{\csname\comma@entry\endcsname}\expandafter{\csname#2\endcsname} } { \ltx@gobble } } %}
} %
\ltx@ifundefined{pdf@strcmp}{%  \zref@delprop
\ZREF@Robust\def\zref@delprop{%  \ZREF@delprop\gdef %}
} %
\ZREF@delprop
417 \def\ZREF@delprop#1#2#3{%
418 \zref@listexists{#2}{%
419 \begingroup
420 \escapechar=-1 %
421 \def\ZREF@param{#3} %
422 \@onelevel@sanitize\ZREF@param
423 \toks@{}%
424 \expandafter\expandafter\expandafter\ZREF@@delprop
425 \csname Z@L@#2\endcsname!%
426 \expandafter\endgroup
427 \expandafter#1\csname Z@L@#2\expandafter\endcsname
428 \expandafter{%
429 \the\toks@
430 }%
431 }%
432 }%

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

\ZREF@delprop
437 \def\ZREF@delprop#1#2#3{%
438 \ifx#2!%
439 \else
440 \def\ZREF@temp{#1}%
441 \@onelevel@sanitize\ZREF@temp
442 \ifx\ZREF@param\ZREF@temp
443 \else
444 \toks0\expandafter{%
445 \the\expandafter\toks0\csname#1\endcsname
446 }%
447 \fi
448 \expandafter\ZREF@delprop
449 \fi
450 }%

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

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

\ZREF@delprop
457 \def\ZREF@delprop#1#2#3{%
458 \zref@listexists{#2}{%
459 \def\ZREF@param{#3} %
460 \edef\ZREF@SavedEscapechar{\the\escapechar}
461 \escapechar=-1 %
462 \expandafter\expandafter\expandafter\ZREF@@delprop
463 \csname Z@L@#2\endcsname
464 }%

31
Caution: #1 might be an \texttt{if} or similar token.

\def\ZREF@@delprop#1{\expandafter\ZREF@@@delprop\expandafter{\string#1}#1\relax}

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

6.2.7 Properties

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

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

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

\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 (\texttt{propname}) is given in \texttt{#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@newprop
\def\ZREF@newprop#1{% 
    \edef\ZREF@P{#1}% 
    \@onelevel@sanitize\ZREF@P 
    \begingroup 
    \ifx\ZREF@P\ZREF@par 
    \PackageError\ZREF@name{Invalid property name \ZREF@P'}{The property name 'par' is not allowed because of internal reasons.}%
    \MessageBreak
    \@ehc 
    \def\ZREF@@newprop[#1]{\endgroup} 
    \else 
    \zref@ifpropundefined\ZREF@P{% 
    \endgroup 
    \PackageInfo\ZREF@name{New property: \ZREF@P}%
    }% 
    \elsePackageError\ZREF@name{Property \ZREF@P' already exists}%
    \@ehc 
    \def\ZREF@newprop[#1]{\endgroup} 
    \fi 
    \@ifnextchar\ZREF@@newprop\ZREF@@newprop[\zref@default]% 
}\ZREF@par
\def\ZREF@par{par} 
\@onelevel@sanitize\ZREF@par 
\ZREF@newprop
\def\ZREF@newprop[#1]{% 
    \global\@namedef{Z@D@\ZREF@P}{#1} 
    \global\expandafter\let\csname Z@X@\ZREF@P\endcsname\ZREF@X 
    \begingroup \expandafter\endgroup \expandafter\ZREF@@@newprop\csname\ZREF@P\endcsname 
    \expandafter\gdef\csname Z@C@\ZREF@P\endcsname{}% 
    \zref@setcurrent\ZREF@P 
}\ZREF@nil
\zref@setcurrent \zref@setcurrent sets the current value for a property.
\ZREF@getcurrent \zref@getcurrent gets the current value for a property.
\ZREF@wu@getcurrent \zref@getcurrent gets the current value for a property.
\ZREF@wu@getcurrent
6.2.8 Reference generation

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

\zref@listexists{#2}%
\expandafter{\expandafter{\expandafter{\zref@label}}}%
\csname Z@L@#2\endcsname{#1}%%
\zref@labelbylist{#1}\ZREF@mainlist

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

\zref@listexists{#2}%
\expandafter\expandafter\expandafter\zref@label{\csname Z@L@#2\endcsname}{#1}%%
\zref@labelbyprops
The properties are directly specified in a comma separated list.

\zref@listexists{#2}%
\expandafter{\expandafter{\expandafter{\zref@label}}}%
\csname Z@L@#2\endcsname{#1}%%
\zref@labelbykv

\zref@labelbykv

\zref@listexists{#2}%
\expandafter{\expandafter{\expandafter{\zref@label}}}%
\csname Z@L@#2\endcsname{#1}%%
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.

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 37is made expandable in this case.
6.2.9 Reference querying and extracting

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

- In standard \LaTeX\ undefined references sets a flag and generate a warning. Both actions are not expandable.

- Babel’s support for its shorthand uses commands that use non-expandable assignments. However currently there is hope, that primitives are added to pdf\LaTeX\ that allows the detection of contexts. Then the shorthand can detect, if they are executed inside \csname and protect themselves automatically.

\zref@ifrefundefined If a reference \#1 is undefined, then macro \zref@ifrefundefined calls \#2 and \#3 otherwise.
\zref@refused The problem with undefined references is addressed by the macro \zref@refused. This can be used outside the expandable context. In case of an undefined reference the flag is set to notify \LaTeX\ and a warning is given.
\ZREF@refused

\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.
6.2.10 Compatibility with \texttt{babel}

\begin{verbatim}
\def\ZREF@def@extractdefault{\edef#1{}}
\end{verbatim}
6.2.11 Unique counter support

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

\begin{verbatim}
\ZREF@Robust\def\zref@require@unique{%
  \ifundefined{c@zref@unique}{%
    \begingroup
    \let\@addtoreset\ltx@gobbletwo
    \newcounter{zref@unique}%
    \endgroup
    \thezref@unique
    \thezref@unique
    is used for automatically generated unique labelnames.
  \renewcommand*{\thezref@unique}{%}
  zref@\number\c@zref@unique
  }%}
\end{verbatim}

6.2.12 Utilities

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

6.2.13 Setup

Standard \LaTeX{} prints "??" in bold face if a reference is not known. \texttt{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 \texttt{ref@newprop}. The global default value can be set by \texttt{zref@setdefault}.

\begin{verbatim}
\ZREF@Robust\def\zref@setdefault#1{%
  \def\zref@default{#1}%
}\zref@setdefault{%
  \fss@text{\reset@font\bfseries ??}%
}\zref@setmainlist
\zref@newlist\ZREF@mainlist
\end{verbatim}

Main property list.

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

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

Now we create the list.

\begin{verbatim}
\zref@newlist\ZREF@mainlist%
\end{verbatim}
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.

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

Properties

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

```latex
\zref@titleref@current Later we will redefine the section and caption macros to catch the current title and remember the value in `\zref@titleref@current`.
```

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

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

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

Mark successful loading

```latex
\let\ZREF@base@ok=Y
```

```latex
6.3 Module user
```

```latex
(*user)
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-user}[2020-05-28 v2.31 Module user for zref (HO)]
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y%
\else
\expandafter\endinput
\fi
```

43
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.

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

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

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

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

\newcommand*{\zref}[2][default]{\zref@propexists{#1}{\zref@wrapper@babel\ZREF@zref{#2}{#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\def{\zref@extract}\zrefused\def{\zref@refused}\

6.4 Module abspage

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-abspage}[2020-05-28 v2.31 Module abspage for zref (HO)]
\RequirePackage{zref-base}[2019/11/29]
Module \texttt{abspage} adds a new property \texttt{abspage} to the main property list for absolute page numbers. These are recorded by the help of package \texttt{atbegshi}.

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

\begin{verbatim}
\begin{group}
\let\@addtoreset\ltx@gobbletwo
\newcounter{abspage}
\end{group}
\setcounter{abspage}{0}
\AtBeginShipout{
\stepcounter{abspage}
\zref@newprop*{abspage}[0]{\the\c@abspage}
\zref@addprop\ZREF@mainlist{abspage}
}
\end{verbatim}

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.

\section{Module \texttt{counter}}

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

\begin{verbatim}
\if\fmtversion{2020-10-01}
\zref@newprop{counter}{\@currentcounter}
\zref@addprop\ZREF@mainlist{counter}
\else
\zref@newprop{counter}{}
\zref@addprop\ZREF@mainlist{counter}
\AtBeginDocument{\ZREF@patch{refstepcounter}{\def\refstepcounter#1{\zref@setcurrent{counter}{#1}\ZREF@org@refstepcounter{#1}}}}
\end{verbatim}
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.

\begin{verbatim}
\zref@newlist{LastPage}
\AfterLastShipout{\if@filesw
\begingroup
\advance\c@page\m@ne
\toks0\expandafter\expandafter\expandafter{\expandafter\Z@L@main\Z@L@LastPage}
\expandafter\zref@wrapper@immediate\expandafter{\expandafter\ZREF@label\expandafter{\the\toks0}{LastPage}}\endgroup\fi}
\zref@iflastpage\def\zref@iflastpage#1{\ifnum\zref@extractdefault{#1}{abspage}{-1}=%
\zref@extractdefault{LastPage}{abspage}{-2} %}
\else\expandafter\ltx@secondoftwo\fi}
\ziflastpage\ZREF@IfDefinable\ziflastpage\def{\zref@wrapper@babel\ZREF@iflastpage}
\end{verbatim}
6.7 Module thepage

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-thepage} \[2020-05-28 v2.31 Module thepage for zref (HO)]
\RequirePackage{zref-base} \[2019/11/29]\ifx\ZREF@base@ok Y\else\expandafter\endinput\fi
\RequirePackage{atbegshi} \[2011/10/05]\RequirePackage{zref-abspage} \[2019/11/29]
\zref@newlist{thepage}
\zref@addprop{thepage}{page}
\zref@addprop{thepage}{pagevalue}
\AtBeginShipout{%
\AtBeginShipoutAddToBox{%
\zref@thepage@atbegshi@hook
\zref@labelbylist{thepage\the\value{abspage}}{thepage}%
}%
}%
\zref@thepage@name
\ltx@IfUndefined{numexpr}{%
\def\zref@thepage@name#1{thepage\number#1}%
}%
\def\zref@thepage@name#1{thepage\the\numexpr#1}%
\zref@thepage
\def\zref@thepage#1{%
\zref@extract{\zref@thepage@name{#1}}{page}%
%}
\zref@thepage@refused
\ZREF@Robust\def\zref@thepage@refused#1{%
\zref@refused{\zref@thepage@name{#1}}%
}%
\zthepage
\ZREF@IfDefinable\zthepage\def{%
#1{%
\zref@thepage@refused{#1}%
\zref@thepage{#1}%
}%
}%
%

6.8 Module nextpage

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-nextpage}[2020-05-28 v2.31 Module nextpage for zref (HO)]
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y
  \else\expandafter\endinput\fi
\RequirePackage{zref-abspage}[2019/11/29]
\RequirePackage{zref-thepage}[2019/11/29]
\RequirePackage{zref-lastpage}[2019/11/29]
\RequirePackage{uniquecounter}[2009/12/18]
\UniqueCounterNew{znextpage}
\newcommand*{\znextpagesetup}{\afterassignment\ZREF@np@setup@i\def\ZREF@np@call@unknown##1\}}
\def\ZREF@np@setup@i{\afterassignment\ZREF@np@setup@ii\def\ZREF@np@call@nonext##1\}}
\def\ZREF@np@setup@ii{\def\ZREF@np@call@next##1\}}
\def\ZREF@np@call@unknown#1{#1}
\def\ZREF@np@call@nonext#1{#1}
\def\ZREF@np@call@next#1{#1}
\ZREF@IfDefinable\znextpage\def{\UniqueCounterCall{znextpage}{\ZREF@nextpage}}
\newcommand*{\znonextpagename}{ }
\newcommand*{\zunknownnextpagename}{\Z@D@page}
\def\ZREF@nextpage#1{\begingroup\def\ZREF@refname@this{zref@np#1}\zref@labelbyprops\ZREF@refname@this{abspage}\chardef\ZREF@call=0 % unknown\ZREF@ifrefundefined\ZREF@refname@this{}{\edef\ZREF@pagenum@this{\zref@extractdefault\ZREF@refname@this{abspage}{0}}\ZREF@ifrefundefined{LastPage}{}{\edef\ZREF@call=2 % next page\chardef\ZREF@call=0 % next page}\ifnum\ZREF@pagenum@this>0 \ZREF@ifrefundefined{LastPage}{}{\edef\ZREF@call=2 % next page\chardef\ZREF@call=0 % next page}\fi\edef\ZREF@call=2 % next page\chardef\ZREF@call=0 % next page}}
\endgroup}
\edef\ZREF@pagenum@last{\zref@extractdefault{LastPage}{abspage}{0} \iffnum\ZREF@pagenum@this<\ZREF@pagenum@last ltx@space \ZREF@ifrefundefined\ZREF@refname@next{% \chardef\ZREF@call=2 % next page \} \else \iffnum\ZREF@pagenum@this=\ZREF@pagenum@this ltx@space \chardef\ZREF@call=1 % no next page \fi \fi \fi \edef\x{\endgroup \ifcase\ZREF@call \noexpand\ZREF@np@call@unknown{% \noexpand\zunknownnextpagename \or \noexpand\ZREF@np@call@nonext{% \noexpand\znonextpagename \else \noexpand\ZREF@np@call@next{% \noexpand\zref@extract{\ZREF@refname@next}{page} \fi \} \x} ⟨/nextpage⟩

\section{Module \texttt{totpages}}

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-totpages} \[2020-05-28 v2.31 Module totpages for zref (HO)]
\RequirePackage{zref-abspage} \[2019/11/29]
\RequirePackage{zref-lastpage} \[2019/11/29]
\ztotpages

Macro \texttt{\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 \texttt{LastPage} as used:
6.10 Module pagelayout

(*pagelayout)
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-pagelayout} [2020-05-28 v2.31 Module pagelayout for zref (HO)]
\RequirePackage{zref-base} [2019/11/29]
\ifx\ZREF@base@ok Y%
  \else
  \expandafter\endinput
\fi
\RequirePackage{zref-thepage} [2019/11/29]
\RequirePackage{iftex} [2019/11/07]
\RequirePackage{atveryend} [2010/03/24]

6.10.1 Define layout properties
\def\ZREF@temp#1{%
\begingroup
\escapechar=-1 %
\ltx@ifundefined{\string#1}{\endgroup}{%
  \edef\x{%
    \endgroup
    \noexpand\zref@newprop*{\string#1}[
    \noexpand\number\noexpand#1]% hash-ok
    {\noexpand\number\noexpand#1}%
    \noexpand\zref@addprop{thepage}{\string#1}%
    }%
\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
\ifluatex
\zref@newprop*{pdfvorigin}%
  [{\number{\pdfvariable vorigin}}] hash-ok
  {\number{\pdfvariable vorigin}}%
\zref@addprop{thepage}{pdfvorigin}%
\zref@newprop*{pdfhorigin}%
  [{\number{\pdfvariable horigin}}] hash-ok
  {\number{\pdfvariable horigin}}%
\zref@addprop{thepage}{pdfhorigin}%
\zref@newprop*{pdfpageheight}%
  [{\number{\pagewidth}}] hash-ok
  {\number{\pagewidth}}%
\zref@addprop{thepage}{pdfpageheight}%
\zref@newprop*{pdfpagewidth}%
  [{\number{\pagewidth}}] hash-ok
  {\number{\pagewidth}}%
\begingroup
\toks@{Page layout parameters:\MessageBreak}%
\count@=1 %
\ZREF@pl@ListPage
\edef\x{\endgroup
\noexpand\PackageInfoNoLine{zref-pagelayout}\{\the\toks@}%
\x}
\ZREF@pl@ListPage
\edef\x{%
\toks@={%\the\toks@
\MessageBreak
\ifnum\ZREF@page@max>\count@%
\advance\count@ by\ltx@one
\else
52
}%
6.11 Module pageattr

(*pageattr*)
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-pageattr} \[2020-05-28 v2.31 Module pageattr for zref (HO)]
\RequirePackage{zref-base} \[2019/11/29] \ifx\ZREF@base@ok Y \else \expandafter\endinput \fi
\RequirePackage{iftex} \[2019/11/07]
\let\ZREF@temp=N \ifluatex \expandafter\@firstoftwo \else \expandafter\@secondoftwo \fi {\luatex}
\RequirePackage{zref-thepage} \[2019/11/29]
\RequirePackage{zref-lastpage} \[2019/11/29]
\zref@newprop*{pdfpageattr}[]{{\zref@hex{\the\pdfvariable pageattr}}}
\zref@addprop{thepage}{pdfpageattr}
\zref@newprop*{pdfpagesattr}[]{{\zref@hex{\the\pdfvariable pagesattr}}}
\zref@addprop{LastPage}{pdfpagesattr}
\let\ZREF@temp=Y \else \expandafter\endinput \fi
\ifluatex \else \fi {\other}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-pageattr} \[2020-05-28 v2.31 Module pageattr for zref (HO)]
\RequirePackage{zref-base} \[2019/11/29] \ifx\ZREF@base@ok Y \else \expandafter\endinput \fi
\RequirePackage{iftex} \[2019/11/07]
\let\ZREF@temp=N \ifluatex \expandafter\@firstoftwo \else \expandafter\@secondoftwo \fi {\luatex}
\RequirePackage{zref-thepage} \[2019/11/29]
\RequirePackage{zref-lastpage} \[2019/11/29]
\zref@newprop*{pdfpageattr}[]{{\zref@hex{\the\pdfvariable pageattr}}}
\zref@addprop{thepage}{pdfpageattr}
\zref@newprop*{pdfpagesattr}[]{{\zref@hex{\the\pdfvariable pagesattr}}}
\zref@addprop{LastPage}{pdfpagesattr}
\let\ZREF@temp=Y \else \expandafter\endinput \fi
\ifluatex \else \fi {\other}
6.12 Module marks

(*marks)
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-marks}[]{2020-05-28 v2.31 Module marks for zref (HO)]
\RequirePackage{zref-base}[2019/11/29]
\ifx\ZREF@base@ok Y\else \expandafter\endinput\fi
\newcommand*{\zref@marks@register}{3\}{\}{%
\let\ZREF@temp=Y\%
\let\ZREF@temp=N\%
\ifluatex \let\ZREF@temp=N \fi
\ifx\ZREF@temp N \expandafter\@firstofone \zref@pdfpagesattr \def\zref@pdfpagesattr{\zref@unhex{\zref@extract{LastPage}{pdfpagesattr}}}%
\zref@pdfpagesattr@used\ZREF@Robust\def\zref@pdfpagesattr@used{\zref@refused{LastPage}%
\zref@pdfpagesattr@used\PackageInfoNoLine{zref-pageattr}{\latexslashchar pdfvariable\else pdf\fi pagesattr:\MessageBreak <<\zref@pdfpagesattr>>\MessageBreak}%
\AfterLastShipout{\ZREF@pa@AfterLastShipout}
\endinput}
\edef\ZREF@TempName{#1}\
\edef\ZREF@TempNum{\ZREF@number{#2}}\
\ifnum\ZREF@TempNum<\ltx@zero\
\PackageError{\ZREF@name}{\string\zref@marks@register\ltx@space is called with invalid marks register number (\ZREF@TempNum)}{Use ‘0’ or the command, defined by \string\newmarks.\MessageBreak @ehc}\
\else\
\ifx\ZREF@TempName\ltx@empty\
\edef\ZREF@TempName{mark\romannumeral\ZREF@TempNum}\
\else\
\edef\ZREF@TempName{marks\ZREF@TempName}\
\fi\
\ZREF@MARKS@DefineProp{top}\
\ZREF@MARKS@DefineProp{first}\
\ZREF@MARKS@DefineProp{bot}\
\kv@parse{#3}{\ifx\kv@value\relax\def\kv@value{top,first,bot}\fi}\
\edef\ZREF@temp{\expandafter\ltx@car\kv@key X\@nil}\
\ifx\ZREF@temp\ZREF@STAR\
\edef\kv@key{\expandafter\ltx@cdr\kv@key\@nil}\
\zref@newlist\kv@key\
\fi\
\expandafter\comma@parse\expandafter{\kv@value}{\ifcase0\ifx\comma@entry\ZREF@NAME@top 1\else\ifx\comma@entry\ZREF@NAME@first 1\else\ifx\comma@entry\ZREF@NAME@bot 1\fi\fi\fi\ltx@space}\
\PackageWarning{\zref-marks}{Use ‘top’, ‘first’ or ‘bot’ for the list values in the third argument of \string\zref@marks@register.\MessageBreak Ignoring unkown value ‘\comma@entry’}\
\else\
\zref@addprop{\kv@key}{\comma@entry\ZREF@TempName}\
\ltx@gobble\
\ltx@gobbletwo\
\fi\
\def\ZREF@STAR{*}\
\def\ZREF@NAME@top{top}\
\def\ZREF@NAME@first{first}\
\def\ZREF@NAME@bot{bot}\
\def\ZREF@MARKS@DefineProp#1{\zref@ifpropundefined{#1\ZREF@TempName}{\ifnum\ZREF@TempNum=\ltx@zero\begingroup\edef\x{\endgroup}}}
6.13 Module runs

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

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-runs} \[2020-05-28 v2.31 Module runs for zref (HO)]

\providecommand*{\zruns}{0}
\AtBeginDocument{\edef\zruns{\number\numexpr\zruns+1}\begingroup\def\on@line{}\PackageInfo{zref-runs}{LaTeX runs: \zruns}\if@filesw \immediate\write\@mainaux{\string\gdef\string\zruns{\zruns}}\fi\endgroup}

6.14 Module perpage

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-perpage} \[2020-05-28 v2.31 Module perpage for zref (HO)]
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. 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{\textbackslash stepcounter}. In fact two nested hooks are used to allow other packages to use the first hook at the beginning of \texttt{\textbackslash stepcounter}.

\texttt{\textbackslash makelperpage} resets a counter at each page break. It uses the same syntax and semantics as \texttt{\textbackslash MakePerPage} from package \texttt{perpage} \cite{perpage}. The initial start value can be given by the optional argument. Default is one that means after the first \texttt{\textbackslash stepcounter} on a new page the counter starts with one.
The heart of this module follows.

First the reference is generated.

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

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

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

Page changes are detected by a changed absolute page number.
\zunmakeperpage Macro \zunmakeperpage cancels the effect of \zmakeperpage.

\zunmakeperpage\fi\endgroup

6.15 Module titleref

6.15.1 Implementation

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

Now we can add the property title is added to the main property list.

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

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

The switch \ifzref@titleref@expand \ZREF@titleref@hook \zref@titleref@cleanup \ZREF@Robust\def\zref@titleref@cleanup#1{"/perpage}
Sometimes a title contains a period at the end. Package `nameref` removes this. This behaviour is controlled by the switch \texttt{\ifzref@titleref@stripperiod} and works regardless of the setting of option `expand`. Period stripping is the default.

\begin{Verbatim}
\ifzref@titleref@stripperiod
\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.

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

\texttt{\ZREF@stripperiod} If \texttt{\ZREF@stripperiod} is called, the argument consists of space tokens and tokens with catcode 12 (other), because of $\varepsilon$-\TeX{}’s \texttt{\detokenize}.

\begin{Verbatim}
\def\ZREF@stripperiod#1\ltx@empty#2\@nil{#1}
\end{Verbatim}

### 6.15.2 User interface

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

\begin{Verbatim}
\define@key{ZREF@TR}{expand}[true]{\csname zref@titleref@expand#1\endcsname}
\define@key{ZREF@TR}{stripperiod}[true]{\csname zref@titleref@stripperiod#1\endcsname}
\define@key{ZREF@TR}{cleanup}{\zref@titleref@cleanup{#1}}
\end{Verbatim}
\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]}%}
  }%
\}%

6.15.7 Package titlesec

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

6.15.8 Package longtable

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

\ifpackageloaded{longtable}{%
  \ZREF@patch{LT@c@ption}{%
    \def\LT@c@ption#1[#2]#3{%
      \ZREF@org@LT@c@ption{#1}{#2}{#3}%
      \zref@titleref@setcurrent{#2}%
    }%
  }%
\}%
6.15.9 Package listings

Package listings: support for its caption.

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

6.15.10 Theorems

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

6.16 Module xr

\NeedsTeXFormat{LaTeX2e} 
\ProvidesPackage{zref-xr} \[2020-05-28 v2.31 Module xr for zref (HO)]% 
\RequirePackage{zref-base} \[2019/11/29]% 
\ifx\ZREF@base@ok Y% 
\else 
\expandafter\endinput 
\fi 
\}%

/*xr*/
We declare property \url, because this is added, if a reference is imported and has not already set this field. Or if \hyperref is used, then this property can be asked.

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

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

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

\newcommand\ZREF@xr@AddURL{\begingroup\def\ZREF@temp{#1}\count@=\ltx@one\ZREF@xr@@AddUrl\endgroup}

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

\def\ZREF@xr@AddURL#1{\begingroup\def\ZREF@temp{#1}\count@=\ltx@one\ZREF@xr@@AddUrl\endgroup}

\def\ZREF@xr@AddURL#1{\begingroup\def\ZREF@temp{#1}\count@=\ltx@one\ZREF@xr@@AddUrl\endgroup}
\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\MessageBreak
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\MessageBreak
\ifZREF@xr@zreflabel zref\else LaTeX\fi\space\MessageBreak
label(s) found\%
\ifnum\ZREF@xr@ignored@empty>0 \MessageBreak
,\MessageBreak\ZREF@xr@ignored@empty\space empty label(s) ignored\%
\fi 
\ifnum\ZREF@xr@ignored@zref>0 \MessageBreak
,\MessageBreak\ZREF@xr@ignored@zref\space duplicated zref label(s) ignored\%
\fi 
\ifnum\ZREF@xr@ignored@ltx>0 \MessageBreak
,\MessageBreak\ZREF@xr@ignored@ltx\space duplicated latex label(s) ignored\%
\fi 
}\%
\endgroup 
}\else 
\edef\ZREF@xr@filelist{\expandafter\ltx@car\ZREF@xr@filelist\@nil}
\edef\ZREF@xr@file{\expandafter\ltx@cdr\ZREF@xr@filelist\ltx@empty\@nil}
\expandafter\ZREF@xr@checkfile
\fi 
\end{verbatim}

\end{verbatim}
The most work must be done for analyzing the arguments of `\newlabel`.
\def\ZREF@xr@ltx@ignorewarning{\PackageWarningNoLine{zref-xr}{\LaTeX\ label `#1' is already in use\MessageBreak in file `\ZREF@xr@file'}\%}
\edef\ZREF@xr@ignored@ltx{\the\numexpr\ZREF@xr@ignored@ltx+1\%}
\ZREF@xr@checklist
\def\ZREF@xr@checklist#1#2#3\ZREF@nil{\ifx\@undefined#1\relax\expandafter\ZREF@xr@checkkey\string#1\@nil\fi\ifx\#3\%\else\ltx@ReturnAfterFi{\ZREF@xr@checklist#3\ZREF@nil}\fi}
\def\ZREF@xr@checkkey#1#2\@nil{\zref@ifpropundefined{#2}{\zref@newprop{#2}{}\%}}
\ZREF@xr@scanparams
\def\ZREF@xr@scanparams#1#2#3#4#5#6#7\ZREF@nil{\let#1\ltx@empty\ZREF@foundfalse\ZREF@xr@scantitleref#1#2\TR@TitleReference{}\ZREF@nil\if\ZREF@found\else\ltx@LocalAppendToMacro#1{\default{#2}}\%\fi\ltx@LocalAppendToMacro#1{\page{#3}}\%\if\ZREF@found\else\ifx\#4\%\else\def\ZREF@xr@temp{\#4}\%\else\def\ZREF@xr@temp{\ZREF@xr@relax}\%\fi\fi\ltx@LocalAppendToMacro#1{\title{\string\ZREF@xr@temp}}\%\ltx@LocalAppendToMacro#1{\anchor{#5\ZREF@xr@temp}}\%\if\ZREF@found\else\fi\ltx@LocalAppendToMacro#1{\anchor{#5\ZREF@temp}}\%\fi\ltx@LocalAppendToMacro#1{\anchor{#6\ZREF@temp}}\%\fi\if\ZREF@found\else\fi\else\fi\endinput
6.17 Module hyperref

UNFINISHED :-(

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-hyperref} [%2020-05-28 v2.31 Module hyperref for zref (HO)]
\RequirePackage{zref-base} [2019/11/29]
\ifx\ZREF@base@ok Y\else \expandafter\endinput \fi
\ZREF@xr@scantitleref
\ZREF@xr@urlcheck

\ZREF@xr@scantitleref
\def\ZREF@xr@scantitleref#1#2\TR@TitleReference#3#4#5\ZREF@nil{% 
\iff\#5\%
\else
\ltx@LocalAppendToMacro#1{
\default{#3} 
\title{#4} 
}%
\ZREF@foundtrue
\fi
\ZREF@xr@urlcheck
\def\ZREF@xr@urlcheck#1{% 
\zref@ifrefcontainsprop{#1}{anchor}{% 
\zref@ifrefcontainsprop{#1}{url}{% 
}\expandafter 
\ltx@LocalAppendToMacro\csname Z@R@#1\expandafter\endcsname
\expandafter{\csname url\ifZREF@xr@urluse use\fi \expandafter\endcsname
\expandafter{\ZREF@xr@url}% 
}%
}%
}%
}%
6.18 Module savepos

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

6.18.1 Identification

\zref@newprop\ZREF@mainlist{anchor}\%
\langle /hyperref \rangle

6.18.2 Availability

First we check, whether the feature is available.
\ifx\directlua\@undefined\ltx@IfUndefined{pdfsavepos}{%\PackageError{\string\pdfsavepos\space is not supported.\MessageBreak It is provided by pdfTeX (1.40) or XeTeX}{\ZREF@UpdatePdfTeX}\expandafter\expandafter\expandafter\endinput\fi

In PDF mode we are done. However support for DVI mode was added later in version 1.40.0. In earlier versions \pdfsavepos is defined, but its execution raises an error. Note that XeTeX also provides \pdfsavepos.
\ifpdf \else\ltx@IfUndefined{pdftexversion}{%\PackageError{\string\pdfsavepos\space is not supported in DVI mode of this pdfTeX version}{\ZREF@UpdatePdfTeX}\expandafter\expandafter\expandafter\endinput\fi

6.18.3 Setup

\zref@newlist{savepos}
\ifx\directlua\@undefined\zref@newprop*{posx}[0]{\the\pdflastxpos}
\zref@newprop*{posy}[0]{\the\pdflastypos}

75
6.18.4 User macros

\zref@savepos

\if\directlua@undefined
\def\zref@savepos{%\if@filesw\pdfsavepos\fi}
\else
\def\zref@savepos{%\if@filesw\savepos\fi}
\fi
\ZREF@zsavepos
\def\ZREF@zsavepos#1#2#3{%\@bsphack\if@filesw\zref@savepos#1{#3}{#2}%%\ltx@IfUndefined{TeXeTstate}{%\ifnum\TeXeTstate=\ltx@zero\else\zref@savepos\fi}%%\fi\@esphack}
\zsavepos The current location is stored in a reference with the given name.
\ZREF@IfDefinable\zsavepos\def{%\ZREF@zsavpos\zref@labelbylist{savepos}%%}
\zsaveposx
\ZREF@IfDefinable\zsaveposx\def{%\ZREF@zsavposx\zref@labelbyprops{posx}%%}
\zsaveposy
\ZREF@IfDefinable\zsaveposy\def{%\ZREF@zsavposy\zref@labelbyprops{posy}%%}
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 PDF\LaTeX. 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.

6.19 Module abspos

6.19.1 Identification

(*)abspos

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-abspos}[2020-05-28 v2.31 Module abspos for zref (HO)]
\RequirePackage{zref-base}[2019/11/29]
\if\ZREF@base@ok
\else
\endinput
\fi
\RequirePackage{zref-savepos}[2019/11/29]
\if\ZREF@savepos@ok
\else
\endinput
\fi
\RequirePackage{zref-pagelayout}[2019/11/29]
\zref@addprop{savepos}{abspage}
\zref@addprop{savepos}{pagevalue}
\zref@absposx
\newcommand*{\zref@absposx}[3]{\number\zref@extractdefault{#1}{abspage}{0}{#2}{#3}\ltxspace}
\zref@absposy
\newcommand*{\zref@absposy}[3]{\number\zref@extractdefault{#1}{abspage}{0}{#2}{#3}\ltxspace}
\zref@absposx
\newcommand*{\zref@absposx}[3]{\number\zref@extractdefault{#1}{abspage}{0}{#2}{#3}\ltxspace}
\zref@absposy
\newcommand*{\zref@absposy}[3]{\number\zref@extractdefault{#1}{abspage}{0}{#2}{#3}\ltxspace}
6.19.2 Media

\ZREF@abspos@media@width
\edef\ZREF@abspos@media@width{%\ifluatex\edef\ZREF@abspos@media@width{pdfpagewidth}\fi}
\ZREF@abspos@media@height
\edef\ZREF@abspos@media@height{%\ifluatex\edef\ZREF@abspos@media@height{pdfpageheight}\fi}
\noexpand ifcase\pageheight
\ifx0\undefined{stockheight}{
\paperheight\}%
\stockheight\}%
\noexpand else
\pdfpageheight\%
\noexpand\fi\%
\fi
\ZREF@abspos@media@x@left
\ZREF@abspos@media@x@right
\ZREF@abspos@media@x@center
\ZREF@abspos@media@y@top
\ZREF@abspos@media@y@bottom
\ZREF@abspos@media@y@center
\ZREF@abspos@paper@x@left
\ZREF@abspos@paper@x@right
\ZREF@abspos@paper@x@center

6.19.3 Paper
6.19.4 Origin

There doesn’t seem a good reason to make these tests depend on pdf mode in current engines, so comment out the \ifpdf tests.
\ZREF@abspos@head@x@left
\ZREF@abspos@head@x@right
\ZREF@abspos@head@x@center
\ZREF@abspos@head@y@top
\ZREF@abspos@head@y@bottom
\def\ZREF@abspos@head@y@bottom#1{%
  \ZREF@abspos@head@y@top(#1)\%
  \textwidth
  -\zref@extract(#1){headheight}\%
}\

\ZREF@abspos@head@y@center
\def\ZREF@abspos@head@y@center#1{%
  \ZREF@abspos@head@y@top(#1)\%
  \textwidth
  -\zref@extract(#1){headheight}/2\%
}\

6.19.6 Body
\ZREF@abspos@body@x@left
\let\ZREF@abspos@body@x@left\ZREF@abspos@head@x@left\
\ZREF@abspos@body@x@right
\let\ZREF@abspos@body@x@right\ZREF@abspos@head@x@right\
\ZREF@abspos@body@x@center
\let\ZREF@abspos@body@x@center\ZREF@abspos@head@x@center\
\ZREF@abspos@body@y@top
\def\ZREF@abspos@body@y@top#1{%
  \ZREF@abspos@head@y@bottom(#1)\%
  \textwidth
  -\zref@extract(#1){headsep}\%
}\
\ZREF@abspos@body@y@bottom
\def\ZREF@abspos@body@y@bottom#1{%
  \ZREF@abspos@body@y@top(#1)\%
  \textwidth
  -\zref@extract(#1){textheight}\%
}\
\ZREF@abspos@body@y@center
\def\ZREF@abspos@body@y@center#1{%
  \ZREF@abspos@body@y@top(#1)\%
  \textwidth
  -\zref@extract(#1){textheight}/2\%
}\

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@foot@y@bottom(#1)\%
  \textwidth
  -\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

6.19.9 Stock paper

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

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

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

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

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

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

85
6.20 Module dotfill

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.

\zdotfill \zdotfill sets labels at the left and the right to get the horizontal position. \savepos is not used, because we do not need the vertical position.
\ZREF@dotfill Help macro that actually sets the dots.
\def\ZREF@dotfill{\cleaders\hb@xt@\dimen@{\hss\ZREF@df@dot\hss}\hfill}

6.21 Module env

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{zref-env} %[2020-05-28 v2.31 Module env for zref (HO)]
\RequirePackage{zref-base} [2019/11/29]
\ifx\ZREF@base@ok Y%
\else
\expandafter\endinput
\fi
\zref@newprop{envname} [\]{\@currenvir}
\zref@newprop{envline} [\]{\zref@env@line}
\zref@env@line
Macro \zref@env@line extracts the line number from \@currenvline.
\def\zref@env@line{\ifx\@currenvline\ltx@empty
\else
\expandafter\ZREF@ENV@line\@currenvline\ltx@empty\ltx@empty\@nil
\fi}
\ZREF@ENV@line
\def\ZREF@ENV@line#1line #2\ltx@empty#3\@nil{#2}

7 Installation

7.1 Download

Package. This package is available on CTAN:2


2CTAN:pkg/zref
**Bundle.** All the packages of the bundle ‘zref’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

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

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

### 7.2 Bundle installation

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

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

### 7.3 Package installation

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

```bash
tex zref.dtx
```

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

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

If you have a docstrip.cfg that configures and enables docstrip’s TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.
7.4 Refresh file name databases

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

7.5 Some details for the interested

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

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

\LaTeX{}: Generate the documentation.

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

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

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

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

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

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

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

8 References

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


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


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


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

9 History

[2006/02/20 v1.0]
- First version.

[2006/05/03 v1.1]
- Module \texttt{perpage} added.
- Module redesign as packages.

[2006/05/25 v1.2]
- Module \texttt{dotfillmin} added.
- Module \texttt{base}: macros \texttt{\zref@require\@unique} and \texttt{\the\zref@unique} added (used by modules \texttt{titleref} and \texttt{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]
- \texttt{\zref@getcurrent} added (suggestion of Igor Akkerman).
- Module \texttt{savepos} also supports \texttt{Xe\TeX}.

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

[2007/04/17 v1.7]
- Package \texttt{atbegshi} replaces \texttt{everyshi}.

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

[2007/05/02 v1.9]
- Module \texttt{titleref}: Some support for \texttt{\caption} of package \texttt{longtable}, but only if \texttt{\label} is given after \texttt{\caption}.
[2007/05/06 v2.0]
- Uses package etexcmds for accessing \LaTeX's \texttt{\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: \texttt{\zref@iflistcontainsprop} is documented, but a broken \texttt{\zref@listcontainsprop} implemented. Name and implementation fixed (thanks Ohad Kammar).

[2008/10/01 v2.3]
- \texttt{\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]
- \texttt{\zifrefundefined} added.
- Module lastpage: Macros \texttt{\zref@iflastpage} and \texttt{\ziflastpage} added.
- Module thepage added.
- Module nextpage added.
[2010/03/29 v2.9]
• Module marks added (without documentation).
• \zref@addprop now adds expanded property to list.
• Useless \ZREF@ErrorNoLine removed.

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

[2010/04/15 v2.11]
• Module titleref: Better support of class memoir.
• Module titleref: Support of theorems.

[2010/04/17 v2.12]
• Module base: \zref@newprop ensures global empty default.
• Module xr: Setup options tozreflabel and toltxlabel added.

[2010/04/19 v2.13]
• \zref@setcurrent throws an error if the property does not exist (Florent Chervet).
• \zref@getcurrent the documentation is fixed (Florent Chervet). Also it returns the empty string in case of errors.
• \zref@addprop and \zref@localaddprop now take a list of property names (feature request of Florent Chervet).
• Example for \zref@wrapper@unexpanded corrected (Florent Chervet).

[2010/04/22 v2.14]
• Bug fix for \zref@getcurrent second argument wasn’t eaten in case of unknown property.
• \zref@getcurrent supports \zref@wrapper@unexpanded.
• \zref@wrapper@unexpanded added for \ZREFxr@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 ‘ltcmds’ and ‘hologo’.
• Module pagelayout added.
• Module pageattr added.
• Robustness introduced for non-expandable interface macros.
• Internal change of the data format of property lists (suggestion of Florent Chervet).
• Module titleref: Support of environment description.

[2010/05/01 v2.17]
• \zref@newprop throws an error if the property already exists.
• Module xr: Bug fix for the case of several .aux files (bug found by Victor Ivrii).
• Module xr: Property ‘urluse’ and option urluse added.

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

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

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

[2011/03/18 v2.21]
• Fix in module pagelayout for \zlistpagelayout.
• Fix for \zref@localaddprop (probably since v2.19).
[2011/10/05 v2.22]

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

[2011/12/05 v2.23]

- Module savepos: \zsaveposx and \zsaveposy added.

[2012/04/04 v2.24]

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

[2016/05/16 v2.25]

- Documentation updates.

[2016/05/21 v2.26]

- update zref-savepos for new luatex

[2018/11/21 v2.27]

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

[2019/11/29 v2.28]

- Documentation updates.
- Use \iftex directly.

[2020-03-03 v2.29]

- adapted in module zref-pagelayout the properties pdfhorigin, pdfvorigin, pdfpagewidth, pdfpageheight for luatex to the right primitives.
- Removed no longer needed code for older lualatex versions.
- added some documentation of the abspos module.
- adapted the abspos module to the new luatex primitives.
- adapted pageattr module to the new luatex primitives.
- added short documentation for pageattr module
- use luatex command names directly in zref-savepos rather than defining pdftex compatibility names.
- allow zref-abspos to use \pdf[vh]origin in dvi mode.
[2020-03-04 v2.30]
• add pagevalue property to savepos in the abspos module (issue 1)

[2020-05-28 v2.31]
• Adapted module zref-counter to use `\@currentcounter` in the next
  \LaTeX version (issue 5)

10 Index

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

### Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\@PackageError</code></td>
<td>508, 524, 2630, 2643</td>
</tr>
<tr>
<td><code>\@PackageInfo</code></td>
<td>2079</td>
</tr>
<tr>
<td><code>\@PackageInfoNoLine</code></td>
<td>548, 563, 1343, 1438, 1450, 1518, 1570</td>
</tr>
<tr>
<td><code>\@PackageWarning</code></td>
<td>691</td>
</tr>
<tr>
<td><code>\@addtoreset</code></td>
<td>911</td>
</tr>
<tr>
<td><code>\@auxout</code></td>
<td>715</td>
</tr>
<tr>
<td><code>\@begintheorem</code></td>
<td>2018, 2023</td>
</tr>
<tr>
<td><code>\@bsphack</code></td>
<td>600, 610, 630, 2502</td>
</tr>
<tr>
<td><code>\@caption</code></td>
<td>1863</td>
</tr>
<tr>
<td><code>\@chapter</code></td>
<td>1875, 1912</td>
</tr>
<tr>
<td><code>\@currentHref</code></td>
<td>940</td>
</tr>
<tr>
<td><code>\@currentcounter</code></td>
<td>1029</td>
</tr>
<tr>
<td><code>\@currentlabel</code></td>
<td>935</td>
</tr>
<tr>
<td><code>\@currenvir</code></td>
<td>2960</td>
</tr>
<tr>
<td><code>\@currenvline</code></td>
<td>2963, 2966</td>
</tr>
<tr>
<td><code>\@esphack</code></td>
<td>607, 627, 642, 2514</td>
</tr>
<tr>
<td><code>\@firstofone</code></td>
<td>1558</td>
</tr>
<tr>
<td><code>\@firstoftwo</code></td>
<td>1910, 1947</td>
</tr>
<tr>
<td><code>\@ifndef</code></td>
<td>242, 289</td>
</tr>
<tr>
<td><code>\@ifnextchar</code></td>
<td>530, 1729</td>
</tr>
<tr>
<td><code>\@ifpackageloaded</code></td>
<td>491, 306, 414, 1578, 2356, 2381, 2391, 2395, 2411</td>
</tr>
<tr>
<td><code>\@iftestdef</code></td>
<td>1325, 1326, 1486, 1487</td>
</tr>
<tr>
<td><code>\@ifundefined</code></td>
<td>1784, 2353, 2456, 2480, 2488</td>
</tr>
<tr>
<td><code>\@mainaux</code></td>
<td>1687</td>
</tr>
<tr>
<td><code>\@namedef</code></td>
<td>535</td>
</tr>
<tr>
<td><code>\@new@bel</code></td>
<td>285</td>
</tr>
<tr>
<td><code>\@nil</code></td>
<td>1617, 1619, 1828, 1835, 2017, 2018, 2171, 2174, 2354, 2363, 2966, 2969</td>
</tr>
<tr>
<td><code>\@onelevel@sanitize</code></td>
<td>422, 440, 505, 533, 2016, 2019</td>
</tr>
<tr>
<td><code>\@opargbegintheorem</code></td>
<td>2008</td>
</tr>
<tr>
<td><code>\@part</code></td>
<td>1869</td>
</tr>
<tr>
<td><code>\@section</code></td>
<td>1893</td>
</tr>
<tr>
<td><code>\@section</code></td>
<td>1899</td>
</tr>
<tr>
<td><code>\@stpl</code></td>
<td>1716, 1717</td>
</tr>
<tr>
<td><code>\@testdef</code></td>
<td>1325, 1326, 1486, 1487</td>
</tr>
<tr>
<td><code>\@testopt</code></td>
<td>2101, 2104, 2114</td>
</tr>
<tr>
<td><code>\@tfor</code></td>
<td>322, 720</td>
</tr>
<tr>
<td><code>\@undefined</code></td>
<td>25, 26, 27, 28, 153, 155, 157, 158, 170, 173, 2262, 2356, 2381, 2391, 2395, 2411</td>
</tr>
<tr>
<td><code>\@AddLineBeginAux</code></td>
<td>280</td>
</tr>
<tr>
<td><code>\@advance</code></td>
<td>1035, 1233, 1681, 1861, 2075, 2924</td>
</tr>
<tr>
<td><code>\@afterassignment</code></td>
<td>233, 1144, 1148</td>
</tr>
<tr>
<td><code>\@AfterLastShipout</code></td>
<td>1057, 1407, 1579</td>
</tr>
<tr>
<td><code>\@Alph</code></td>
<td>7</td>
</tr>
<tr>
<td><code>\@anchor</code></td>
<td>2393</td>
</tr>
<tr>
<td><code>\@AtBeginDocument</code></td>
<td>1035, 1233, 1681, 1861</td>
</tr>
<tr>
<td><code>\@AtBeginShipout</code></td>
<td>1012, 1104</td>
</tr>
<tr>
<td><code>\@AtBeginShipoutAddToBox</code></td>
<td>1105</td>
</tr>
<tr>
<td><code>\@AtBeginShipoutBoxDepth</code></td>
<td>1315</td>
</tr>
<tr>
<td><code>\@AtBeginShipoutBoxHeight</code></td>
<td>1314</td>
</tr>
<tr>
<td><code>\@AtBeginShipoutBoxWidth</code></td>
<td>1313</td>
</tr>
<tr>
<td><code>\@AtEndOfPackage</code></td>
<td>195</td>
</tr>
<tr>
<td><code>\@AtVeryEndDocument</code></td>
<td>1334, 1495</td>
</tr>
<tr>
<td><code>\@beamer@section</code></td>
<td>1949</td>
</tr>
<tr>
<td><code>\@beamer@subsection</code></td>
<td>1955</td>
</tr>
<tr>
<td><code>\@beamer@subsubsection</code></td>
<td>1961</td>
</tr>
<tr>
<td><code>\@begin</code></td>
<td>23, 57, 100, 106, 156, 172</td>
</tr>
<tr>
<td><code>\@bfseries</code></td>
<td>928</td>
</tr>
</tbody>
</table>

A

B

