

# The undolabl package\*

H.-Martin Münch  
(Martin dot Muench at Uni-Bonn dot de)

2010/06/03

## Abstract

This package allows to override existing labels, especially automatically generated ones.

Note: The main code of this package was invented and published on the internet by **Ulrich Diez** ([eu.angelion@web.de](mailto:eu.angelion@web.de)) in 2008. While Ulrich Diez neither wanted to create a package himself nor have one published under his name, he granted the publication of his code. Therefore: Thanks! I submitted this package to CTAN (after some updates of the documentation, mainly layout, typos and such things.)

Disclaimer for web links: The author is not responsible for any contents referred to in this work unless he has full knowledge of illegal contents. If any damage occurs by the use of information presented there, only the author of the respective pages might be liable, not the one who has referred to these pages.

Save per page about 200 ml water, 2 g CO<sub>2</sub> and 2 g wood: Therefore please print only if this is really necessary.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Usage</b>	<b>4</b>
<b>3</b>	<b>Example</b>	<b>5</b>
<b>4</b>	<b>The implementation</b>	<b>6</b>
<b>5</b>	<b>Installation</b>	<b>8</b>
5.1	Downloads . . . . .	8
5.2	Package, unpacking TDS . . . . .	9
5.3	Refresh file name databases . . . . .	9
5.4	Some details for the interested . . . . .	9
<b>6</b>	<b>Acknowledgements</b>	<b>10</b>
<b>7</b>	<b>History</b>	<b>10</b>
	[2008/04/20 v0.3] . . . . .	10
	[2010/04/08 v0.3b] . . . . .	10
	[2010/06/01 v1.0] . . . . .	10
	[2010/06/03 v1.0b] . . . . .	10
<b>8</b>	<b>Index</b>	<b>11</b>

---

\*This file has version number v1.0b, last revised 2010/06/03, documentation dated 2010/06/03.

# 1 Introduction

The package's name `undolabl` is an eight-letter abbreviation for the phrases “undo” and “label”.

This package allows to override existing labels, especially automatically generated ones.

When an instance of the `\label`-macro occurs in the document (e.g. `\label{foo}`), then a delayed `\write` (a write which is performed at shipout-time when the page-number is determined) to the `.aux` file is issued and you find in the `.aux` file something like:

```
\newlabel{foo}{<page-no>{<sectional no>}{<probably something else>}}
```

During the beginning of the next  $\text{\LaTeX}$ -run, the `.aux` file will be read and

```
\newlabel{foo}{<page-no>{<sectional no>}{<probably something else>}}
```

is expanded to

```
\@newl@bel r{foo}{<page-no>{<sectional no>}{<probably something else>}}
```

which in turn gets expanded to something like (pseudo code):

```
IF (macro \r@foo is already defined)
THEN
  - Issue an error-message,
  - Make sure that the multiply-defined-labels-warning occurs
    in the log-file
ELSE
  \def\r@foo{<page-no>{<sectional no>}{<probably something else>}}
ENDIF
```

Referencing works as follows:

`\ref{foo}` ->

- expand `\r@foo`: `{<page-no>{<sectional no>}{<probably something else>}}`
- grab the second from the resulting arguments: `<sectional no>`

`\pageref{foo}` ->

- expand `\r@foo`: `{<page-no>{<sectional no>}{<probably something else>}}`
- grab the first from the resulting arguments: `<page-no>`

Back to the `\label`-mechanism:

- At the beginning of the  $\text{\LaTeX}$ -run, all the `\r@<label>`-macros get defined from reading the `.aux` file. The `\r@<label>`-macros get used by the referencing-macros (`\ref`, `\pageref`...) during the  $\text{\LaTeX}$ -run.
- During the  $\text{\LaTeX}$ -run, the `.aux` file gets rewritten.
- At the end of the  $\text{\LaTeX}$ -run, the `.aux` file (which was rewritten/newly created during the  $\text{\LaTeX}$ -run) is read in order to detect whether references have changed during the current  $\text{\LaTeX}$ -run.

But this time `\@newl@bel` is redefined (`\let equal to \@testdef`) and thus this time

```
\@newl@bel r{foo}{<page-no>{<sectional no>}{<probably something else>}}
```

expands to something like (pseudo code):

```
Compare the (newly written) third argument (that is:
{{<page-no>}}<sectional no>}}<probably something else>}} )
to the (current/former) definition of \r@foo.
If the two are different, then some page- or section-number
related to referencing has changed from the last to the current
LaTeX-run, thus in this case issue a message in the log-file:
"References may have changed. Rerun LaTeX in order to get
cross-references right".
```

So what do you need to do in order to override a label:

- First you need to write to the .aux file to silently undefine the associated \r@<label>-macro if it is already defined. That's why \overridelabel writes in terms of \protected@write to the .aux file:

```
\undonewlabel{<label>}{<code-line-number>}
```

\undonewlabel "undefines" the \r@<label>-macro. (How this works will be explained below.)

- Then \overridelabel can call \label{<label>} again and thus produce another \newlabel{<label>} -entry to the .aux file.

In the .aux file all this results in a sequence like:

```
% from the former \label-call:
\newlabel{<label>}{..}
|->| \r@<label>-macro gets produced.
% from the \overridelabel-call:
% - call to \undonewlabel within \overridelabel:
\undonewlabel{<label>}{<code-line-number>}
|->| \r@<label>-macro gets destroyed.
% - call to \label within \overridelabel:
\newlabel{<label>}{..}
|->| a new \r@<label>-macro and no multiply-label-defined warning
gets produced.
```

There is another issue left:

- It was said that .aux file is read at the beginning and at the end of the LaTeX-run for detecting whether references have changed.
- When overriding a <label>, there will be several \newlabel{<label>} -calls associated to the same label-name in the .aux file.
- At the beginning of the LaTeX-run only the last one counts for defining the associated \r@<label>-macro.
- But at the end, when the new .aux file is read, they all count and thus with all these entries but the last one, the above-mentioned \@testdef-comparison will yield difference and thus in any case cause a warning-message about references having changed although that might not be a correct statement.

Ulrich Diez decided to catch this up by letting \undonewlabel "count" within the macro \s@<label> how many times the label was overridden. That means: Whenever \r@<label> is overridden, the definition of \s@<label> gets a character "i" attached.

The \@testdef-comparison-mechanism gets enhanced via replacing<sup>1</sup> it by \undolabl@testdef: Before actually executing \@testdef, check whether

---

<sup>1</sup> "replacing" works as follows: When .aux file is read at the beginning of the LaTeX-run, newl@bel is not let equal to testdef. When .aux file is read at the end of the LaTeX-run, newl@bel is let equal to testdef. Thus it is sufficient to write into the beginning of the .aux file a direction which leads to letting newl@bel equal to \my@testdef in case its definition equals testdef. That direction is called "\reset@newl@bel". Also, when the .aux file is read at the end of the LaTeX-run, \undonewlabel-entries therein should do nothing, thus \undonewlabel is let equal to gobbletwo.

`\s@<label>` is defined and empty. If undefined or empty, perform `\@testdef`. If not empty, remove an "i" from its definition-text. This way it is ensured that only the last one of several `\newlabel{<label>}`-entries is used for the `\@testdef`-comparison.

```

\undonewlabel{<label>}{<code-line-number>}
->
\@und@newl@bel rs{<label>}{<code-line-number>}
->
% IF \r@<label> is undefined just \relax, otherwise:
\@ifundefined{r@<label>}\relax{%
  % Inform via |.log| file about overriding the label:
  \@latex@warning@no@line{Label '<label>' overridden<code-line-number>}%
  % Undefine \r@label via letting it equal to \relax:
  \expandafter\global
  \expandafter\let
  \csname r@<label>\endcsname\relax
  % IF \s@label is undefined, define it to {i} ELSE
  % add another i to it via \g@addto@macro ENDIF:
  \@ifundefined{s@<label>}\global\@namedef{s@<label>}{i}}{%
  \expandafter\g@addto@macro\csname s@<label>\endcsname{i}%
}%
}%

```

## 2 Usage

Load the package placing

```
\usepackage{undolabl}
```

in the preamble of your L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> source file.

When an existing label shall be replaced by a new one, say `\overridelabel{<label name>}` (where `<label name>` is the name of the label to be replaced by the new one), instead of just `\label{<label name>}`, which would produce a LaTeX Warning: Label '`<label name>`' multiply defined.

### 3 Example

```
1 (*example)
2 \documentclass[british]{article}
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 \usepackage{undolabl}[2010/06/03]% v1.0b
5 %% There are no options for the undolabl package. %%
6 \usepackage[hyperref]{2010/05/13}% v6.81e; if you want to use nameref
7 \hypersetup{%
8   extension=pdf,%
9   plainpages=false,%
10  pdfpagelabels=true,%
11  hyperindex=false,%
12  pdflang={en},%
13  pdftitle={undolabl package example},%
14  pdfauthor={Hans-Martin Muench},%
15  pdfsubject={Example for the undolabl package},%
16  pdfkeywords={LaTeX, undolabl, Hans-Martin Muench},%
17  pdfview=FitH,%
18  pdfstartview=FitH,%
19  pdfpagelayout=OneColumn,%
20  bookmarksopen=true%
21 }
22 \makeatletter
23 %% Code from tcilatex.tex, Macros for Scientific Word and Scientific WorkPlace 5.5 <06 Oct 20
24 %% Copyright (C) 2005 Mackichan Software, Inc.
25 %% That macro file is NOT proprietary and may be freely copied and distributed.
26 \def\unit#1{\mathord{\thinspace\rm #1}}%
27 %% End of code from tcilatex.tex
28 \makeatother
29 \listfiles
30 \begin{document}
31 \pagenumbering{arabic}
32 \section*{Example for undolabl}
33
34 This example demonstrates the use of package\newline
35 \textsf{undolabl}, v1.0b as of 2010/06/03 (HMM).\newline
36 The used options were \texttt{linktextcolour=black},\newline
37 \texttt{backgroundcolour=white}, and \texttt{pdfborder={0 0 1}}\newline
38 (the default ones).\newline
39 For more details please see the documentation!\newline
40
41 \noindent Save per page about $200\unit{ml}$ water,\newline
42 $2\unit{g}$ CO$_2$ and $2\unit{g}$ wood:\newline
43 Therefore please print only if this is really necessary.\newline
44
45 \bigskip
46
47 \section{Test}
48
49 text \label{testlabel}\label{testlabel}
50 page-reference: \pageref{testlabel}\label{testlabel} -> page 4
51 sectional-reference: \ref{testlabel}\label{testlabel} -> section 4
52 name-reference: \nameref{testlabel}\label{testlabel} -> Still another test
53
54 \newpage
55
56 \section{Another test}
57
58 text \overridelabel{testlabel}\label{testlabel}
59 page-reference: \pageref{testlabel}\label{testlabel} -> page 4
60 sectional-reference: \ref{testlabel}\label{testlabel} -> section 4
61 name-reference: \nameref{testlabel}\label{testlabel} -> Still another test
62
63 \newpage
64
65 \section{Yet another test}
66
```

```

67 text \overridelabel{testlabel}\\
68 page-reference: \pageref{testlabel}\\% -> page 4
69 sectional-reference: \ref{testlabel}\\% -> section 4
70 name-reference: \nameref{testlabel}% -> Still another test
71
72 \newpage
73
74 \section{Still another test}
75
76 text \overridelabel{testlabel}\\
77 page-reference: \pageref{testlabel}\\% -> page 4
78 sectional-reference: \ref{testlabel}\\% -> section 4
79 name-reference: \nameref{testlabel}% -> Still another test
80
81 \end{document}
82 </example>

```

## 4 The implementation

For a somewhat longer description see section 1.

We start off by checking that we are loading into L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and announcing the name and version of this package.

```

83 (*package)
84 \NeedsTeXFormat{LaTeX2e}[1994/06/01]
85 \ProvidesPackage{undolabl}[2010/06/03 v1.0b
86   Overriding labels (HMM)]
87

```

`\overridelabel` In order to override a label, first you need to write to the .aux file to silently undefine the associated `\r@<label>-macro` if it is already defined. That's why `\overridelabel` writes in terms of `\protected@write` to the .aux file: `\undonewlabel{<label>}{<code-line-number>}`, "undefining" the `\r@<label>-macro`.

```

88 \newcommand\overridelabel[1]{%
89   \bsphack
90   \protected@write\auxout{}{\string\undonewlabel{#1}{\on@line}}%

```

Then `\overridelabel` can call `\label{<label>}` again and thus produce another `\newlabel{<label>}`-entry to the .aux file.

```

91   \label{#1}%
92   \esphack
93 }%
94

```

`\undonewlabel`

```

95 \newcommand\undonewlabel{\@und@newl@bel rs}%
96

```

`\@und@newl@bel`

If `\r@<label>` is undefined, just `\relax`

```

97 \newcommand\@und@newl@bel[4]{%
98   \ifundefined{#1@#3}\relax{%

```

otherwise:

Inform via .log file about overriding the label:

```

99   \@latex@warning@no@line{Label ‘#3’ overridden#4}%

```

Undefine `\r@label` via letting it equal to `\relax`:

```

100   \expandafter\global
101   \expandafter\let
102   \csname #1@#3\endcsname\relax

```

```

        If \s@label is undefined, define it to {i}
103     \ifundefined{#2@#3}{\global\@namedef{#2@#3}{i}}{%
        else add another i to it via \g@addto@macro.

104     \expandafter\g@addto@macro\csname #2@#3\endcsname{i}%
105     }%
106     }%
107     }%
108

\undolabl@testdef

109 \newcommand\undolabl@testdef[3]{%
110   \ifundefined{s@#2}\@secondoftwo\@firstofone{%
111     \expandafter\ifx\csname s@#2\endcsname\empty
112     \expandafter\@firstofone
113   \else
114     \expandafter\xdef\csname s@#2\endcsname{%
115       \expandafter\expandafter
116       \expandafter\@gobble
117       \csname s@#2\endcsname
118     }%
119     \expandafter\@gobble
120     \fi
121   }%
122   {%
123     \@testdef{#1}{#2}{#3}%
124   }%
125   }%
126

\protected@write
  @auxout 127 \protected@write\@auxout{}\string\reset@newl@bel}%
128

\reset@newl@bel

129 \newcommand\reset@newl@bel{%
130   \ifx\@newl@bel\@testdef
131     \let\@newl@bel\undolabl@testdef
132     \let\undonewlabel\@gobbletwo
133   \fi
134   }%
135 \end{package}

```

## 5 Installation

### 5.1 Downloads

Everything *should* be available on CTAN: <ftp://ftp.ctan.org/tex-archive/>, but may need additional packages themselves.

`undolabl.dtx` For unpacking the `undolabl.dtx` file and constructing the documentation it is required:

- T<sub>E</sub>XFormat L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, 1994/06/01, v2<sub>ε</sub>: [CTAN:](#)
- document class `ltxdoc`, 2007/11/11, v2.0u,  
[CTAN:macros/latex/base/ltxdoc.dtx](#)
- package `hltxdoc`, 2010/04/18, v0.18,  
[CTAN:macros/latex/contrib/oberdiek/hltxdoc.dtx](#)
- package `hypdoc`, 2010/03/26, v1.9,  
[CTAN:macros/latex/contrib/oberdiek/hypdoc.dtx](#)

`undolabl.sty` The `undolabl.sty` for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (i. e. all documents using the `undolabl` package) requires:

- T<sub>E</sub>XFormat L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, 1994/06/01, v2<sub>ε</sub>, [CTAN:](#)

`undolabl-example.tex` The `undolabl-example.tex` requires the same files as all documents using the `undolabl` package, and additionally:

- class `article`, 2007/10/19, v1.4h, from `classes.dtx`:  
[CTAN:macros/latex/base/classes.dtx](#)
- package `hyperref`, 2010/05/13, v6.81,  
[CTAN:macros/latex/contrib/hyperref/](#),  
if `nameref` shall be used  
(and when the references shall be hyperlinked, of course)
- package `undolabl`, 2010/06/03, v1.0b,  
[CTAN:macros/latex/contrib/muench/undolabl/undolabl.dtx](#)  
(submitted, should become available at that location soon)  
(Well, it is the example file for this package, and because you are reading the documentation for the `undolabl` package, it can be assumed that you already have some version of it – is it the current one?)

`Oberdiek` All packages of Heiko Oberdiek’s bundle ‘`oberdiek`’ (especially `hltxdoc` and  
`hltxdoc` `hypdoc`) are also available in a TDS compliant ZIP archive:

`hypdoc` [CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#).

**Warning:** `hltxdoc`, 2010/04/24 v0.19, requires the packages

- `hypdoc`, 2010/03/26, v1.9
- `hyperref`, 2010/03/30, v6.80u (latest: 2010/05/13, v6.81e)
- `pdftexcmds`, 2010/04/01, v0.9
- `ltxcmds`, 2010/03/09, v1.4 (latest: 2010/04/26, v1.7)
- `hologo`, 2010/04/24, v1.2
- `array` (latest: 2008/09/09, v2.4c)

(or more recent versions) and does neither work with nor check for earlier versions!  
(It is probably best to download

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#) and use this, because the packages in there should be both recent and compatible.)



## 5.2 Package, unpacking TDS

**Package.** This package should become available on CTAN: soon:

CTAN:macros/latex/contrib/muench/undolabl/undolabl.dtx

The source file.

CTAN:macros/latex/contrib/muench/undolabl/undolabl.ins

The installation file.

CTAN:macros/latex/contrib/muench/undolabl/undolabl.drv

The driver to generate the documentation.

CTAN:macros/latex/contrib/muench/undolabl/ltxdoc.cfg

The L<sup>A</sup>T<sub>E</sub>X documentation configuration file, also for generating the documentation.

CTAN:macros/latex/contrib/muench/undolabl/undolabl.pdf

The documentation.

CTAN:macros/latex/contrib/muench/undolabl/undolabl.sty

The **style** file.

CTAN:macros/latex/contrib/muench/undolabl/undolabl-example.tex

The example file.

CTAN:macros/latex/contrib/muench/undolabl/undolabl-example.pdf

The compiled example file, as it should look like.

CTAN:install/macros/latex/contrib/muench/undolabl/undolabl.tds.zip

Everything in TDS compliant, compiled format.

For required other packages, see the preceding subsection.

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain T<sub>E</sub>X:

```
tex undolabl.dtx
```

About generating the documentation see paragraph 5.4 below.

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

<code>undolabl.sty</code>	→ <code>tex/latex/muench/undolabl.sty</code>
<code>undolabl.pdf</code>	→ <code>doc/latex/muench/undolabl.pdf</code>
<code>undolabl-example.tex</code>	→ <code>doc/latex/muench/undolabl-example.tex</code>
<code>undolabl-example.pdf</code>	→ <code>doc/latex/muench/undolabl-example.pdf</code>
<code>undolabl.dtx</code>	→ <code>source/latex/muench/undolabl.dtx</code>

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

## 5.3 Refresh file name databases

If your T<sub>E</sub>X distribution (teT<sub>E</sub>X, miK<sub>T</sub>E<sub>X</sub>,...) relies on file name databases, you must refresh these. For example, teT<sub>E</sub>X users run `texhash` or `mktextlsr`.

## 5.4 Some details for the interested

**Unpacking with L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain T<sub>E</sub>X:** Run docstrip and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for docstrip (really, docstrip does not need L<sup>A</sup>T<sub>E</sub>X), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{undolabl.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 the following line into this file, if you want to have A4 as paper format:

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

An example follows how to generate the documentation with pdfL<sup>A</sup>T<sub>E</sub>X:

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

and with pdfL<sup>A</sup>T<sub>E</sub>X:

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

## 6 Acknowledgements

I (H.-Martin Münch) would like to thank **Ulrich Diez** ([eu.angelion@web.de](mailto:eu.angelion@web.de)) for writing the main code of this package as well as for the permission to publish it on **CTAN**. I also thank those anonymous people who had published it on the internet. Further I would like to thank Heiko Oberdiek ([heiko dot oberdiek at googlemail dot com](mailto:heiko_dot_oberdiek_at_googlemail_dot_com)) for providing a lot (!) of usefull packages (from which I also got everything I know about creating a file in `dtx` format, ok, say it: copying), and the [news:comp.text.tex](#) and [news:de.comp.text.tex](#) newsgroups for their help in all things T<sub>E</sub>X.

## 7 History

[2008/04/20 v0.3]

- created by Ulrich Diez

[2010/04/08 v0.3b]

- `.dtx` updated by H.-Martin Münch, submitted to **CTAN**: (no changes in the style code).

[2010/06/01 v1.0]

- `.dtx` updated by H.-Martin Münch: some minor corrections in the documentation, an internal renaming for possible better compatibility with other packages.
- The main code of this package was invented in 2008 by Ulrich Diez ([eu.angelion@web.de](mailto:eu.angelion@web.de)) and published on the internet. Because Ulrich Diez neither wanted to create a package himself nor have one published under his name, but granted the publication of his code (Thanks!), I had to change author/maintainer of this package and resubmit it.

[2010/06/03 v1.0b]

- Found an unchanged reference to the package author/maintainer.
- Example adapted to other examples of mine.
- Updated references to other packages.
- TDS locations updated.
- Several changes in the documentation and the Readme file.

## 8 Index

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

Symbols			
<code>\@auxout</code> .....	90, 127	<code>\listfiles</code> .....	29
<code>\@bspack</code> .....	89	<b>M</b>	
<code>\@espack</code> .....	92	<code>\mathord</code> .....	26
<code>\@firstofone</code> .....	110, 112	<b>N</b>	
<code>\@gobble</code> .....	116, 119	<code>\nameref</code> .....	52, 61, 70, 79
<code>\@gobbletwo</code> .....	132	<code>\newcommand</code> .....	88, 95, 97, 109, 129
<code>\@ifundefined</code> .....	98, 103, 110	<b>O</b>	
<code>\@latex@warning@no@line</code> .....	99	<code>\Oberdiek</code> .....	8
<code>\@namedef</code> .....	103	<code>\on@line</code> .....	90
<code>\@newl@bel</code> .....	130, 131	<code>\overridelabel</code> .....	6, 58, 67, 76, 88
<code>\@secondoftwo</code> .....	110	<b>P</b>	
<code>\@testdef</code> .....	123, 130	<code>\pagenumbering</code> .....	31
<code>\@und@newl@bel</code> .....	6, 95, 97	<code>\protected@write</code> .....	90, 127
<b>C</b>		<code>\protected@write\@auxout</code> .....	7
<code>\csname</code> .....	102, 104, 111, 114, 117	<code>\ProvidesPackage</code> .....	85
<b>E</b>		<b>R</b>	
<code>\empty</code> .....	111	<code>\ref</code> .....	51, 60, 69, 78
<code>\endcsname</code> ...	102, 104, 111, 114, 117	<code>\reset@newl@bel</code> .....	7, 127, 129
<b>G</b>		<code>\rm</code> .....	26
<code>\g@addto@macro</code> .....	104	<b>U</b>	
<b>H</b>		<code>\undolabl-example.tex</code> .....	8
<code>\holtxdoc</code> .....	8	<code>\undolabl.dtx</code> .....	8
<code>\hypdoc</code> .....	8	<code>\undolabl.sty</code> .....	8
<code>\hypersetup</code> .....	7	<code>\undolabl@testdef</code> .....	7, 109, 131
<b>L</b>		<code>\undonewlabel</code> .....	6, 90, 95, 132
<code>\label</code> .....	49, 91	<code>\unit</code> .....	26, 41, 42