

# The *changes*-package

Manual change markup — version 3.0.0

November 4, 2018

**Ekkart Kleinod**

✉ [ekleinod@edgesoft.de](mailto:ekleinod@edgesoft.de)

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Using the <i>changes</i>-package</b>	<b>4</b>
2.1	Available scripts . . . . .	6
<b>3</b>	<b>Limitations and possible enhancements</b>	<b>7</b>
<b>4</b>	<b>User interface of the <i>changes</i>-package</b>	<b>8</b>
4.1	Package Options . . . . .	8
4.1.1	draft . . . . .	8
4.1.2	final . . . . .	8
4.1.3	markup . . . . .	9
4.1.4	addedmarkup, deletedmarkup . . . . .	9
4.1.5	highlightmarkup . . . . .	10
4.1.6	commentmarkup . . . . .	11
4.1.7	authormarkup . . . . .	11
4.1.8	authormarkupposition . . . . .	12
4.1.9	authormarkuptext . . . . .	12
4.1.10	todonotes . . . . .	13
4.1.11	truncate . . . . .	13
4.1.12	ulem . . . . .	13
4.1.13	xcolor . . . . .	14
4.2	Change management . . . . .	14
4.2.1	\added . . . . .	14
4.2.2	\deleted . . . . .	15
4.2.3	\replaced . . . . .	15
4.3	Highlighting and Comments . . . . .	15
4.3.1	\highlight . . . . .	15
4.3.2	\comment . . . . .	16
4.4	Overview of changes . . . . .	16
4.4.1	\listofchanges . . . . .	16
4.5	Author management . . . . .	17
4.5.1	\definechangesauthor . . . . .	17
4.6	Adaptation of the output . . . . .	18
4.6.1	\setaddedmarkup . . . . .	18
4.6.2	\setdeletedmarkup . . . . .	18
4.6.3	\sethighlightmarkup . . . . .	19
4.6.4	\setcommentmarkup . . . . .	19
4.6.5	\setauthormarkup . . . . .	19
4.6.6	\setauthormarkupposition . . . . .	20
4.6.7	\setauthormarkuptext . . . . .	20
4.6.8	\settruncatewidth . . . . .	20

4.6.9	\setsummarywidth . . . . .	21
4.6.10	\setsummarytewidth . . . . .	21
4.6.11	\setsoceextension . . . . .	21
4.7	Used packages . . . . .	22
<b>5</b>	<b>Authors</b>	<b>23</b>
<b>6</b>	<b>Versions</b>	<b>24</b>
<b>7</b>	<b>Distribution, Copyright, License</b>	<b>25</b>

## 1 Introduction

This package provides means for manual change markup.

Any comments, thoughts or improvements are welcome. The package is maintained at *gitlab*, please see

<http://edgesoft.de/projects/latex/changes/>

for links to source code access, bug and feature tracker, etc. If you want to contact me directly, please send an email to [ekleinod@edgesoft.de](mailto:ekleinod@edgesoft.de). Please start your email subject with [changes].

README: The changes-package allows the user to manually markup changes of text, such as additions, deletions, or replacements. Changed text is shown in a different color; deleted text is striked out. Additionally, text can be highlighted and/or commented. The package allows free definition of additional authors and their associated color. It also allows you to change the markup of changes, authors, highlights or comments.

Here is a short example of change markup:

This is **new**<sup>EK</sup> text. In this sentence, I replace a ~~goodbad~~<sup>[EK 1]: a good word is better than a bad one</sup> word. And, to sum up the text changes, there is another ~~obsolete~~<sup>EK</sup> word to delete. Furthermore, text can be **highlighted**<sup>EK</sup> or just <sup>[EK 2]: For the fun of it.</sup> commented.

## 2 Using the *changes*-package

In this section a typical use case of the *changes*-package is described. You can find the detailed description of the package options and new commands in Section 4.

We start with the text you want to change. You want to markup the changes for each author individually. Such a change markup is well-known in WYSIWYG text processors such as *LibreOffice*, *OpenOffice*, or *Word*.

The *changes*-package was developed in order to support such change markup. The package provides commands for defining authors, and for marking text as added, deleted, or replaced. Additionally, text can be highlighted or commented. In order to use the package, you have to follow these steps:

1. use *changes*-package
2. define authors
3. markup text changes
4. highlight and comment text
5. typeset the document with  $\text{\LaTeX}$
6. output list of changes
7. remove markup

### use *changes*-package

In order to activate change management, use the *changes*-package as follows:

```
\usepackage{changes}
```

respectively

```
\usepackage[<options>]{changes}
```

You can use the options for defining the layout of the change markup. You can change the layout after using the *changes*-package as well.

For detailed information please refer to Section 4.1 and Section 4.6.

### define authors

The *changes*-package provides a default anonymous author. If you want to track your changes depending on the author, you have to define the needed authors as follows:

```
\definechangesauthor[<options>]{<id>}
```

Every author is uniquely identified through his or her id. You can give every author an optional name and/or color.

For detailed information please refer to Section 4.5.

### markup text changes

Now everything is set to markup the changed text. Please use the following commands according to your change:

for newly added text:

```
\added[id=<id>, comment=<comment>]{new text}
```

for deleted text:

```
\deleted[id=<id>, comment=<comment>]{old text}
```

for replaced text:

```
\replaced[id=<id>, comment=<comment>]{<new text>}{<old text>}
```

Stating the author's id and/or a comment is optional.

For detailed information please refer to Section 4.2.

### highlight and comment text

Maybe you want to highlight or comment some text?

highlight text:

```
\highlight[id=<id>, comment=<comment>]{text}
```

comment text:

```
\comment[id=<id>]{comment}
```

Stating the author's id and/or a comment for highlights is optional.

For detailed information please refer to Section 4.3.

### typeset the document with $\LaTeX$

After marking your changes in the text you are able to display them in the generated document by processing it as usual with  $\LaTeX$ . By processing your document the changed text is layouted as you stated by the corresponding options and/or special commands.

### output list of changes

You can print a list of changes using:

```
\listofchanges[style=<list|summary|compactsummary>, title=<title>,  
show=<type>]
```

The list is meant to be the analogon to the list of tables, or the list of figures.

Stating the style is optional, default is style=list. In order to print a quick overview of the number and kind of changes of every author, use the option style=summary or style=compactsummary. Show only specific changes by using the show option.

By running  $\TeX$  the data of the list is written into an auxiliary file. This data is used in the next  $\TeX$  run for typesetting the list of changes. Therefore, two  $\TeX$  runs are needed after every change in order to typeset an up-to-date list of changes.

For detailed information please refer to Section 4.4.

### **remove markup**

Often you want to remove the change markup after acknowledging or rejecting the changes. You can suppress the output of changes with:

```
\usepackage[final]{changes}
```

## **2.1 Available scripts**

In order to remove the markup from the  $\TeX$  source code you can use a script from Silvano Chiaradonna. You find the script in the directory:

```
<texpath>/scripts/changes/
```

The script removes all markups. You can select or deselect markup from removal using the interactive mode. Switch on the interactive mode with the script parameter `-i`.

The script is not adapted yet to the comment and highlight commands.

### 3 Limitations and possible enhancements

The *changes*-package was carefully programmed and tested. Yet the possibility of errors in the package exists, you might encounter problem during use, or you might miss functionality. In that case, please go to

<http://changes.sourceforge.net/>

There you can report errors, ask for help in the forum, or give advice to other users. You can view the source code, and change it according to your needs. I will try to include your changes in the maintained package. If you are a registered *sourceforge* user you can be a co-author of the *changes*-package.

You can write me an email too, please send it to [ekleinod@edgesoft.de](mailto:ekleinod@edgesoft.de). In that case, please start your email subject with [changes].

Change markup of texts works well, it is possible to markup whole paragraphs. You can markup more than one paragraph at a time but occasionally this leads to errors. You cannot markup figures or tables.

You can try putting such text in an extra file and include in with `input`. This works sometimes, give it a try. Kudos to Charly Arenz for this tip.

There is a problem of typesetting footnotes or margin notes in special environments, such as tables or tabblings. Avoid such markup when using these environments.

There are several possibilities of enhancing the *changes*-package. I will describe but a few here, I will not implement them due to lack of time and/or skill. You can have a look at the more complete list of enhancements on the *gitlab* page.

- selecting of acknowledged and rejected texts; deletion of the corresponding markup
- markup of more than one paragraph
- markup of figures and tables
- automatic markup based on diff information (with regard to the limitations, such as markup of paragraphs, figures etc.)
- translation of language dependent texts and the user documentation in other languages



## 4 User interface of the *changes*-package

This section describes the user interface of the *changes*-package, i.e. all options and commands of the package. Every option respectively new command is described. If you want to see the options and commands in action, please refer to the examples in

`<texpath>/doc/latex/changes/examples/`

The example files are named with the used option respectively command.

A preliminary remark regarding typesetting of replaced text: replaced text is always typeset as follows: `<new text><old text>`. Thus, there is no possibility to influence the output of replaced text directly, but via changing the output of added respectively deleted text.

### 4.1 Package Options

#### 4.1.1 **draft**

The *draft*-option enables markup of changes. The list of changes is available via `\listofchanges`. This option is the default option, if no other option is selected.

The *changes* package reuses the declaration of *draft* in `\documentclass`. The local declaration of *final* overrules the declaration of *draft* in `\documentclass`.

`\usepackage[draft]{changes} \equiv \usepackage{changes}`

#### 4.1.2 **final**

The *final*-option disables markup of changes, only the correct text will be shown. The list of changes is disabled, too.

The *changes* package reuses the declaration of *final* in `\documentclass`. The local declaration of *draft* overrules the declaration of *final* in `\documentclass`.

`\usepackage[final]{changes}`

### 4.1.3 markup

The markup option chooses a predefined visual markup of changed text. The default markup is chosen if no explicit markup is given. The markup chosen with markup can be overwritten with the more special markup options addedmarkup, deletedmarkup, commentmarkup, or highlightmarkup.

The following values are allowed:

**default**

default markup for added and deleted text, comments and highlighted text (default markup)

**underlined**

underlined for added text, wavy underlined for highlighted text, default for deleted text, and comments

**bfit**

bold added text, italic deleted text, default for comments and highlighted text

**nocolor**

no colored markup, underlined for added text, wavy underlined for highlighted text, default for deleted text and comments

*Call*

```
\usepackage[markup=<markup>]{changes}
```

*Examples*

```
\usepackage[markup=default]{changes} ≡ \usepackage{changes}
```

```
\usepackage[markup=underlined]{changes}
```

```
\usepackage[markup=bfit]{changes}
```

```
\usepackage[markup=nocolor]{changes}
```

When changing from color markup to markup without color and vice versa, some errors occur if an auxiliary file exists. Please ignore the errors, they vanish in the second run.

### 4.1.4 addedmarkup, deletedmarkup

The addedmarkup option chooses a predefined visual markup of added text. The deletedmarkup option chooses a predefined visual markup of deleted text respectively. The default markup is chosen if no explicit markup is given. The options addedmarkup and deletedmarkup overwrite the markup chosen with markup.

The following values are allowed:

**colored**

no text markup, just coloring – **example** (default markup for added text)

**uline**

underlined text – example

**uuline**  
double underlined text – example

**uwave**  
wavy underlined text – example

**dashuline**  
dashed underlined text – example

**dotuline**  
dotted underlined text – example

**sout**  
striked out text (deleted text only) – ~~example~~ (default markup for deleted text)

**xout**  
crossed out text (deleted text only) – ~~example~~

**bf** bold text – **example**

**it** italic text – *example*

**sl** slanted text – *example*

**em** emphasized text – *example*

#### Call

`\usepackage[addedmarkup=markup]{changes}`

#### Examples

`\usepackage[addedmarkup=colored]{changes} \usepackage{changes}`

`\usepackage[addedmarkup=uuline]{changes}`

#### Call

`\usepackage[deletedmarkup=markup]{changes}`

#### Examples

`\usepackage[deletedmarkup=sout]{changes} \usepackage{changes}`

`\usepackage[deletedmarkup=xout]{changes}`

`\usepackage[deletedmarkup=uwave]{changes}`

### 4.1.5 highlightmarkup

The `highlightmarkup` option chooses a predefined visual markup for highlighted text. The default markup is chosen if no explicit markup is given. The option `highlightmarkup` overwrites the markup chosen with `markup`.

The following values are allowed:

**background**  
markup by background color – example (default markup for highlighted text)

**uuline**  
double underlined text – example

**uwave**  
wavy underlined text – example

*Call*

```
\usepackage[highlightmarkup=markup]{changes}
```

*Examples*

```
\usepackage[highlightmarkup=background]{changes} ≡ \usepackage{changes}
\usepackage[highlightmarkup=uwave]{changes}
```

#### 4.1.6 commentmarkup

The `commentmarkup` option chooses a predefined visual markup for comments. The default markup is chosen if no explicit markup is given. The option `commentmarkup` overwrites the markup chosen with `markup`.

The following values are allowed:

`todo`

comment as todo note, which is not added to list of todos – (default markup for comments)

`margin`

comment in margin –

`footnote`

comment as footnote – <sup>1</sup>

`uwave`

wavy underlined text – example comment

*Call*

```
\usepackage[commentmarkup=markup]{changes}
```

*Examples*

```
\usepackage[commentmarkup=todo]{changes} ≡ \usepackage{changes}
\usepackage[commentmarkup=footnote]{changes}
```

#### 4.1.7 authormarkup

The `authormarkup` option chooses a predefined visual markup of the author's identification. The default markup is chosen if no explicit markup is given.

The following values are allowed:

`superscript`

superscripted text – text<sup>author</sup> (default markup)

`subscript`

subscripted text – text<sub>author</sub>

---

<sup>1</sup> example comment

brackets

text in brackets – text(author)

footnote

text in footnote – text<sup>2</sup>

none

no author identification

*Call*

```
\usepackage[authormarkup=<markup>]{changes}
```

*Examples*

```
\usepackage[authormarkup=superscript]{changes} ≡ \usepackage{changes}
```

```
\usepackage[authormarkup=subscript]{changes}
```

```
\usepackage[authormarkup=brackets]{changes}
```

```
\usepackage[authormarkup=footnote]{changes}
```

```
\usepackage[authormarkup=none]{changes}
```

#### 4.1.8 authormarkupposition

The `authormarkupposition` option chooses the position of the author's identification. The default value is chosen if no explicit markup is given.

The following values are allowed:

right

right of the text – text<sup>example</sup> (default value)

left

left of the text – <sup>example</sup>text

*Call*

```
\usepackage[authormarkupposition=<markup>]{changes}
```

*Examples*

```
\usepackage[authormarkupposition=right]{changes} ≡ \usepackage{changes}
```

```
\usepackage[authormarkupposition=left]{changes}
```

#### 4.1.9 authormarkuptext

The `authormarkuptext` option chooses the text that is used for the author's identification. The default value is chosen if no explicit markup is given.

The following values are allowed:

id author's id – text<sup>id</sup> (default value)

name

author's name – text<sup>authorname</sup>

---

<sup>2</sup> author

*Call*

```
\usepackage[authormarkuptext=<markup>]{changes}
```

*Examples*

```
\usepackage[authormarkuptext=id]{changes}  $\cong$  \usepackage{changes}  
\usepackage[authormarkuptext=name]{changes}
```

#### 4.1.10 todonotes

Options for the *todonotes* package can be specified as parameters of the *todonotes*-option. Two or more options have to be put in curly brackets.

*Call*

```
\usepackage[todonotes=<options>]{changes}
```

*Examples*

```
\usepackage[todonotes={textsize=tiny}]{changes}
```

#### 4.1.11 truncate

Options for the *truncate* package can be specified as parameters of the *truncate*-option. Two or more options have to be put in curly brackets.

*Call*

```
\usepackage[truncate=<options>]{changes}
```

*Examples*

```
\usepackage[truncate=hyphenate]{changes}
```

#### 4.1.12 ulem

Options for the *ulem* package can be specified as parameters of the *ulem*-option. Two or more options have to be put in curly brackets.

*Call*

```
\usepackage[ulem=<options>]{changes}
```

*Examples*

```
\usepackage[ulem=UWforbf]{changes}  
\usepackage[ulem={normalem,normalbf}]{changes}
```

### 4.1.13 xcolor

Options for the *xcolor* package can be specified as parameters of the `xcolor`-option. Two or more option have to be embraced in curly brackets.

*Call*

```
\usepackage[xcolor=<options>]{changes}
```

*Examples*

```
\usepackage[xcolor=dvipdf]{changes}
```

```
\usepackage[xcolor={dvipdf,gray}]{changes}
```

## 4.2 Change management

All examples use comment markup *uwave*, because todo notes or margin notes are not allowed in tabbing environments (see Section 3).

### 4.2.1 \added

`\added`

The command `\added` marks new text. The new text is the mandatory argument for the command, thus it is written in curly braces.

The optional argument contains key-value-pairs for author-id and comment. The author-id has to be defined using `\definechangesauthor`. If the comment contains special characters or spaces, use curly brackets to enclose the comment.

If a comment is given, the direct author markup at the changes text is omitted, because the author is printed in the comment.

*Call*

```
\added[id=<author's id>, comment=<comment>]{<new text>}
```

*Examples*

This is `\added[id=EK]{new}` text.

This is `newEK` text.

This is `\added[id=EK, comment={has to be in it}]{new}` text.

This is `new[EK 3]: has to be in it` text.

This is `\added[comment=anonymous]{new}` text.

This is `new[1]: anonymous` text.

### 4.2.2 `\deleted`

`\deleted`

The command `\deleted` marks deleted text. For arguments see `\added`.

*Call*

```
\deleted[id=<author's id>, comment=<comment>]{<deleted text>}
```

*Examples*

This is `\deleted[comment=obsolete]{bad}` text.

This is `bad[2]; obsolete` text.

### 4.2.3 `\replaced`

`\replaced`

The command `\replaced` marks replaced text. Mandatory arguments are the new text and the old text. For optional arguments see `\added`.

*Call*

```
\replaced[id=<author's id>, comment=<comment>]{<new text>}{<old text>}
```

*Examples*

This is `\replaced[id=EK]{nice}{bad}` text.

This is `nicebadEK` text.

## 4.3 Highlighting and Comments

### 4.3.1 `\highlight`

`\highlight`

The command `\highlight` highlights text and adds a comment to the document. The highlighted text is the mandatory argument for the command, thus it is written in curly braces.

The optional argument contains key-value-pairs for author-id and comment. The author-id has to be defined using `\definechangesauthor`. If the comment contains special characters or spaces, use curly brackets to enclose the comment.

If a comment is given, the direct author markup at the changes text is omitted, because the author is printed in the comment.



*Call*

```
\highlight[id=<author's id>, comment=<comment>]{<highlighted text>}
```

*Examples*

This is \highlight[id=EK, comment={my comment}]{highlighted} text.

This is highlighted [\[EK 4\]: my comment](#) text.

This is \highlight[comment={anonymous comment}]{anonymously highlighted} text.

This is anonymously highlighted [\[3\]: anonymous comment](#) text.

### 4.3.2 \comment

`\comment`

The command `\comment` adds a comment to the document. The comment is the mandatory argument for the command, thus it is written in curly braces.

The optional argument contains a key-value-pair for the author-id. The author-id has to be defined using `\definechangesauthor`.

The comments are numbered, the number is printed in the comment.

*Call*

```
\comment[id=<author's id>]{<comment>}
```

*Examples*

This is commented\comment[id=EK]{my comment} text.

This is commented [\[EK 5\]: my comment](#) text.

This is commented\comment{anonymous comment} text.

This is commented [\[4\]: anonymous comment](#) text.

## 4.4 Overview of changes

### 4.4.1 \listofchanges

`\listofchanges`

The command `\listofchanges` outputs a list or summary of changes. The first  $\TeX$ -run creates an auxiliary file, the second run uses the data of this file. Therefore you need two  $\TeX$ -runs for an up-to-date list of changes.

There are three optional arguments:

- style
- title
- show

**style** The style argument defines the layout of the list of changes. There are three styles allowed:

- `list` – prints the list of changes like a list of figures (default)
- `summary` – prints the number of changes grouped by author
- `compactsummary` – same as `summary` but entries with count 0 are omitted

**title** The title argument is used to change the title for the list. If you want to use special characters or spaces in the title, enclose it in curly braces.

**show** The show argument defines which types of change markup are shown in the list of changes. The following values are allowed:

- `all` – show all types (default)
- `added` – show only additions
- `deleted` – show only deletions
- `replaced` – show only replacements
- `highlight` – show only highlights
- `comment` – show only comments

You can combine the values using the `|` character. For example if you want to show all additions and deletions, use `show=added|deleted`.

*Call*

```
\listofchanges[style=<list|summary|compactsummary>, title=<title>,  
show=<type>]
```

*Examples*

```
\listofchanges  
\listofchanges[style=list] ≙ \listofchanges  
\listofchanges[style=summary, title={My Summary}]  
\listofchanges[title={List of comments}, show=comment]
```

## 4.5 Author management

### 4.5.1 `\definechangesauthor`

`\definechangesauthor`

The command `\definechangesauthor` defines a new author for changes. You have to define a unique author's id, special characters or spaces are not allowed within the author's id. You may define a corresponding color and the author's name. If you do not define a color, blue is used. The author's name is used in the list of changes and in the markup, if you set the corresponding option.

*Call*

```
\definechangesauthor[name={\langle author's name \rangle}, color={\langle color \rangle}]{\langle author's id \rangle}
```

*Examples*

```
\definechangesauthor{EK}  
\definechangesauthor[color=orange]{EK}  
\definechangesauthor[name={Ekkart Kleinod}]{EK}  
\definechangesauthor[name={Ekkart Kleinod}, color=orange]{EK}
```

## 4.6 Adaptation of the output

### 4.6.1 `\setaddedmarkup`

`\setaddedmarkup`

The command `\setaddedmarkup` defines the layout of added text. The default markup is colored text, or the markup set with the option `markup` respectively `addedmarkup`.

Values for definition: any  $\TeX$ -commands, added text can be used with “#1”.

*Call*

```
\setaddedmarkup{\langle definition \rangle}
```

*Examples*

```
\setaddedmarkup{\emph{#1}}  
\setaddedmarkup{+++ : #1}
```

### 4.6.2 `\setdeletedmarkup`

`\setdeletedmarkup`

The command `\setdeletedmarkup` defines the layout of deleted text. The default markup is striked-out, or the markup set with the option `markup` respectively `deletedmarkup`.

Values for definition: any  $\TeX$ -commands, deleted text can be used with “#1”.

*Call*

```
\setdeletedmarkup{\langle definition \rangle}
```

*Examples*

```
\setdeletedmarkup{\emph{#1}}  
\setdeletedmarkup{--- : #1}
```

### 4.6.3 `\sethighlightmarkup`

`\sethighlightmarkup`

The command `\sethighlightmarkup` defines the layout of highlighted text. The default markup is via a background color, or the markup set with the option `markup` respectively `highlightmarkup`.

Values for definition: any  $\TeX$ -commands, highlighted text can be used with “#1”. The author’s color can be used with “authorcolor”.

*Call*

```
\sethighlightmarkup{<definition>}
```

*Examples*

```
\sethighlightmarkup{\emph{#1}}
```

```
\sethighlightmarkup{\ifthenelse{\isColored}{\color{authorcolor!60}}{!!: #1}}
```

### 4.6.4 `\setcommentmarkup`

`\setcommentmarkup`

The command `\setcommentmarkup` defines the layout of comments. The default markup is a margin note, or the markup set with the option `markup` respectively `commentmarkup`.

Values for definition: any  $\TeX$ -commands, comment text can be used with “#1”, author’s id with “#2”, and author output (id or name) with “#3”. The author’s color can be used with “authorcolor” and the comment count of the autor is stored in “authorcommentcount”.

*Call*

```
\setcommentmarkup{<definition>}
```

*Examples*

```
\setcommentmarkup{-- #1 --}
```

```
\setcommentmarkup{\ifthenelse{\isColored}{\color{authorcolor}}{}}{#1}
```

```
\setcommentmarkup{\ifthenelse{\isAnonymous{#2}}{\textbf{#3: }}{#1}}
```

```
\setcommentmarkup{[\arabic{authorcommentcount}] #1}
```

### 4.6.5 `\setauthormarkup`

`\setauthormarkup`

The command `\setauthormarkup` defines the layout of the author’s markup in the text. The default markup is a superscripted author’s text.

Values for definition: any  $\TeX$ -commands, author’s text can be used with “#1”.

*Call*

```
\setauthormarkup{<definition>}
```

*Examples*

```
\setauthormarkup{(#1)}  
\setauthormarkup{(#1)~---~}  
\setauthormarkup{\marginpar{#1}}
```

**4.6.6 \setauthormarkupposition**

`\setauthormarkupposition`

The command `\setauthormarkupposition` defines the position of the author's markup relative to the changed text. The default position is right of the changed text.

Possible values: *left* == left of the changes; all other values: right

*Call*

```
\setauthormarkupposition{<position>}
```

*Examples*

```
\setauthormarkupposition{left}
```

**4.6.7 \setauthormarkuptext**

`\setauthormarkuptext`

The command `\setauthormarkuptext` defines the text for the author's markup. The default markup is the author's id.

Possible values: *name* == author's name; all other values: author's id

*Call*

```
\setauthormarkuptext{<text>}
```

*Examples*

```
\setauthormarkuptext{name}
```

**4.6.8 \settruncatewidth**

`\settruncatewidth`

The command `\settruncatewidth` sets the width of the truncation in the list of changes to the given value. The default width is `0.6\textwidth`.

```
\settruncatewidth{<length>}
```

```
\settruncatewidth{5cm}  
\settruncatewidth{.3\textwidth}
```

#### 4.6.9 `\setsummarywidth`

```
\setsummarywidth
```

The command `\setsummarywidth` sets the width of the list of changes in summary style to the given length. The default width is `0.3\textwidth`.

```
\setsummarywidth{<length>}
```

```
\setsummarywidth{3cm}  
\setsummarywidth{.5\textwidth}
```

#### 4.6.10 `\setsummarytewidth`

```
\setsummarytewidth
```

The command `\setsummarytewidth` sets the width of the list of changes in summary style to the width of the given text.

```
\setsummarytewidth{<text>}
```

```
\setsummarytewidth{Highlighted \qqquad}  
\setsummarytewidth{The longest text you can imagine for the summary.}
```

#### 4.6.11 `\setsocextension`

```
\setsocextension
```

The command `\setsocextension` sets the extension of the auxiliary file for the summary of changes (soc-file<sup>3</sup>). The default extension is “soc”. In the example stated below, the soc-file for “foo.tex” would be named “foo.changes” instead of the default name “foo.soc”.

*Call*

```
\setsocextension{<extension>}
```

*Examples*

```
\setsocextension{changes}
```

---

<sup>3</sup> “soc” stands for “summary of changes”.

## 4.7 Used packages

The *changes*-package uses already existing packages for its functions. You will find detailed description of the packages in their distributions.

The following packages are always required and have to be installed for the *changes*-package:

**xifthen**

provides an enhanced `\if`-command as well as a `while`-loop

**xkeyval**

provides options with key-value-pairs

**xstring**

improves string operations

The following packages are sometimes required and have to be installed if used by the corresponding option:

**pdfcolmk**

loaded if colored text is used for markup (default markup); solves the problem of colored text and page breaks (with `pdflatex`)

**todonotes**

loaded if comments are layouted as todo notes (default markup)

**ulem**

loaded if text has to be striked or exed out (default markup)

**xcolor**

loaded if colored text is used for markup (default markup)

## 5 Authors

Several authors contributed to the *changes*-package. Many bugs and problems were solved or their solution inspired via de.comp.text.tex. Thanks.

The authors are (in alphabetical order):

- Chiaradonna, Silvano
- Fischer, Ulrike
- Giovannini, Daniele
- Kleinod, Ekkart
- Mittelbach, Frank
- Voss, Herbert
- Wölfel, Philipp
- Wolter, Steve



## 6 Versions

For a list of versions and the changes within these version, please refer to

<https://gitlab.com/ekleinod/changes/blob/master/changelog.md>

Here you too find the implemented but not released changes for the new version.

If you are interested in planned new features, please see

<https://gitlab.com/ekleinod/changes/milestones>

## 7 Distribution, Copyright, License

Copyright 2007-2018 Ekkart Kleinod (ekleinod@edgesoft.de)

This work may be distributed and/or modified under the conditions of the  $\text{\LaTeX}$  Project Public License, either version 1.3 of this license or any later version. The latest version of this license is in <http://www.latex-project.org/lppl.txt> and version 1.3 or later is part of all distributions of  $\text{\LaTeX}$  version 2005/12/01 or later.

This work has the LPPL maintenance status “maintained”. The current maintainer of this work is Ekkart Kleinod.

This work consists of the files

```
source/latex/changes/changes.drv
source/latex/changes/changes.dtx
source/latex/changes/changes.ins
source/latex/changes/examples.dtx
source/latex/changes/README
source/latex/changes/userdoc/*.tex
scripts/changes/delcmdchanges.bash
```

and the derived files

```
doc/latex/changes/changes.english.pdf
doc/latex/changes/changes.english.withcode.pdf
doc/latex/changes/changes.ngerman.pdf
doc/latex/changes/examples/changes.example.*.tex
doc/latex/changes/examples/changes.example.*.pdf
tex/latex/changes/changes.sty
```