

Das *changes*-Paket

Manuelle Änderungsmarkierung – Version 0.6.0

Ekkart Kleinod
ekleinod@edgesoft.de

January 9, 2012

Contents

1	Einleitung	3
2	Benutzung des <i>changes</i>-Pakets	3
3	Einschränkungen und Erweiterungsmöglichkeiten	5
4	Die Benutzerschnittstelle des <i>changes</i>-Pakets	6
4.1	Paketoptionen	7
4.1.1	draft	7
4.1.2	final	7
4.1.3	markup	7
4.1.4	addedmarkup, deletedmarkup	8
4.1.5	authormarkup	9
4.1.6	authormarkupposition	9
4.1.7	authormarkuptext	10
4.1.8	ulem	10
4.1.9	xcolor	10
4.2	Änderungsmanagement	11
4.2.1	\added	11
4.2.2	\deleted	11
4.2.3	\replaced	11
4.2.4	\listofchanges	12
4.3	Autorenverwaltung	12
4.3.1	\definechangesauthor	12
4.4	Anpassung der Ausgabe	13
4.4.1	\setaddedmarkup	13
4.4.2	\setdeletedmarkup	13
4.4.3	\setauthormarkup	13
4.4.4	\setauthormarkupposition	14
4.4.5	\setauthormarkuptext	14
4.4.6	\setremarkmarkup	15
4.4.7	\setlocextension	15
4.5	Sonstige neue Befehle	15
4.5.1	\textsubscript	15
4.6	Benötigte Pakete	16
5	Autoren	16
6	Versionen	17
7	Weitergabe, Copyright, Lizenz	19

8	The documented sourcecode	19
8.1	Package information and options	20
8.1.1	Package options	20
8.1.2	Command options	24
8.1.3	Option processing	24
8.2	Packages	25
8.3	Language dependent texts	25
8.4	File extension	26
8.5	Authors	27
8.5.1	Author management	27
8.5.2	Author markup	28
8.6	Change management commands	29
8.6.1	Text markup definition	29
8.6.2	Change management command definition	30
8.7	List of changes	33

1 Einleitung

Dieses Paket dient dazu, manuelle Änderungsmarkierung zu ermöglichen.

Verbesserungsvorschläge, Gedanken oder Kritik sind willkommen. Das Paket wird auf *sourceforge* gehalten, bitte gehen Sie zu

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

für Quellcodezugang, Fehler- und Featuretracker, Forum etc. Wenn Sie mich direkt kontaktieren wollen, mailen Sie bitte an ekleinod@edgesoft.de . Bitte starten Sie Ihr Mail-Subject mit [changes].

README: Das changes-Paket dient zur manuellen Markierung von geänderten Text, insbesondere Einfügungen, Löschungen und Ersetzungen. Der geänderte Text wird farbig markiert und, bei gelöschtem Text, durchgestrichen. Das Paket ermöglicht die freie Definition von Autoren und deren zugeordneten Farben. Es erlaubt zusätzlich die Änderung des Änderungs-, Autor- und Anmerkungsmarkups.

2 Benutzung des *changes*-Pakets

In diesem Kapitel wird die Nutzung des *changes*-Pakets beschrieben. Dabei wird ein typischer Anwendungsfall geschildert. Die ausführliche Beschreibung der Paketoptionen und neuen Befehle finden Sie nicht hier, sondern in [section 4](#).

Ausgangslage ist ein Text, an dem Änderungen vorgenommen werden sollen. Diese Änderungen sollen markiert werden, und zwar für jeden Autor einzeln. Eine solche Änderungsmarkierung ist z. B. von WYSIWYG-Textprogrammen wie *LibreOffice*, *OpenOffice* oder *Word* bekannt.

Zu diesem Zweck wurde das *changes*-Paket entwickelt. Das Paket stellt Befehle zur Verfügung, um verschiedene Autoren zu definieren und Text als zugefügt, gelöscht oder geändert zu markieren. Um das Paket zu nutzen, müssen Sie folgende Schritte ausführen:

1. *changes*-Paket einbinden
2. Autoren definieren
3. Textänderungen markieren
4. Dokument mit \LaTeX setzen
5. Liste von Änderungen anzeigen lassen
6. Markierungen entfernen

***changes*-Paket einbinden**

Um die Änderungsverfolgung zu aktivieren, ist das *changes*-Paket wie folgt einzubinden:

```
\usepackage{changes}
```

bzw.

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

Mit den verfügbaren Optionen bestimmen Sie hauptsächlich das Aussehen der Änderungsmarkierungen. Sie können das Aussehen der Änderungsmarkierungen auch nach Einbinden des *changes*-Pakets verändern.

Für Details lesen Sie bitte [section 4.1](#) und [section 4.4](#).

Autoren definieren

Das *changes*-Paket stellt einen vordefinierten anonymen Autor zur Verfügung. Wenn Sie jedoch die Änderungen per AutorIn verfolgen wollen, müssen Sie die entsprechenden AutorInnen definieren. Dies geht wie folgt:

```
\definechangesauthor[<optionen>]{ID}
```

Über die ID werden der/die AutorIn und die zugehörigen Textänderungen eindeutig identifiziert. Optional können Sie einen Namen angeben und dem/der AutorIn eine eigene Farbe zuweisen.

Für Details lesen Sie bitte [section 4.3](#).

Textänderungen markieren

Jetzt ist alles vorbereitet, um den geänderten Text zu markieren. Benutzen Sie bitte je nach Änderung die folgenden Befehle:

für neu zugefügten Text:

```
\added[<ID>][<Anmerkung>]{Text}
```

für gelöschten Text:

```
\deleted[<ID>][<Anmerkung>]{Text}
```

für geänderten Text:

```
\replaced[<ID>][<Anmerkung>]{neuer Text}{alter Text}
```

Die Angabe von Autoren-ID und einer Anmerkung ist optional.

Für Details lesen Sie bitte [section 4.2](#).

Dokument mit \LaTeX setzen

Nachdem Sie die Änderungen im \LaTeX -Text markiert haben, können Sie sie im erzeugten Dokument sichtbar machen, indem Sie das Dokument ganz normal übersetzen. Durch die Übersetzung wird der geänderte Text so markiert, wie Sie das mittels der Optionen bzw. speziellen Befehle eingestellt haben.

Liste von Änderungen anzeigen lassen

Sie können sich eine Liste der Änderungen ausgeben lassen. Dies erfolgt mit dem Kommando:

```
\listofchanges
```

Die Ausgabe ist gedacht als Analogon zur Liste von Tabellen oder Abbildungen. Sie dient dazu, einen schnellen Überblick über Art und Anzahl der Änderungen abhängig von dem/der AutorIn zu bekommen.

Bei jedem \LaTeX -Lauf werden die Daten für diese Liste in eine Hilfsdatei geschrieben. Beim nächsten \LaTeX -Lauf werden dann diese Daten genutzt, um die Änderungsliste anzuzeigen. Daher sind nach jeder Änderung zwei \LaTeX -Läufe notwendig, um eine aktuelle Änderungsliste anzuzeigen.

Markierungen entfernen

Oft ist es der Fall, dass die Änderungen eines Dokuments angenommen oder abgelehnt werden und nach diesem Prozess die Änderungsmarkierungen entfernt werden sollen. Sie können die Ausgabe der Änderungsmarkierungen per Option beim Einbinden des *changes*-Pakets unterdrücken:

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

Für die Entfernung der Markierungen aus dem Quelltext steht ein Script von Silvano Chiaradonna zur Verfügung. Das Script liegt im Verzeichnis:

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

Das Script entfernt alle Markierungen, es ist nicht möglich, die zu entfernenden Markierungen zu selektieren.

3 Einschränkungen und Erweiterungsmöglichkeiten

Das *changes*-Paket ist sorgfältig programmiert und getestet worden. Dennoch kann es vorkommen, dass Fehler im Paket sind, dass die Benutzung problematisch ist oder dass eine Funktion fehlt, die Sie gerne hätten. In diesem Fall gehen Sie bitte zu

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

Dort können Sie Fehler melden, im Forum um Hilfe fragen oder Tips einstellen. Sie können dort den Quellcode ansehen und nach Ihren Wünschen ändern bzw. erweitern. Ich werde mich dann bemühen, Ihre Änderungen einzuarbeiten. Sie können als Co-Autor am Paket mitarbeiten, wenn Sie bei *sourceforge* angemeldet sind.

Sie können mir auch eine Mail schreiben an ekleinod@edgesoft.de, in diesem Fall starten Sie bitte Ihr Mail-Subject mit [changes].

Die Änderungsmarkierung von Text funktioniert recht gut, es können auch ganze Absätze markiert werden. Die Markierung von mehreren Absätzen gleichzeitig, von Bildern und Tabellen ist nicht möglich.

Fußnoten (die standardmäßige Auszeichnung von Anmerkungen) werden in bestimmten Umgebungen, z. B. Tabellen oder der *tabbing*-Umgebung, nicht korrekt gesetzt, dort erscheinen also Anmerkungen nicht. Das kann gelöst werden, indem eine andere Annotation von Anmerkungen definiert wird.

Das Paket bietet Raum für Erweiterungen, die ich jedoch nicht selbst programmieren werde (weil mir Zeit und oft auch die Fähigkeit fehlt). Ich liste hier einige Möglichkeiten auf, eine komplettere Liste finden Sie auf der *sourceforge*-Seite:

- Auswahl der anzunehmenden/abzulehnenden Änderungen mit entsprechendem Löschen des Textes
- Markierung von mehreren Absätzen
- Markierung von Bildern und Tabellen
- automatische Markierung anhand von diff-Informationen (unter Berücksichtigung der Einschränkungen bzgl. Absätzen, Bildern, etc.)
- Übersetzung der sprachabhängigen Texte und der Nutzerdokumentation in andere Sprachen

4 Die Benutzerschnittstelle des *changes*-Pakets

In diesem Kapitel wird die Nutzerschnittstelle des *changes*-Pakets vorgestellt, d. h. alle Optionen und Kommandos. Jede Option bzw. jedes neue Kommando werden beschrieben. Wenn Sie die Optionen und Kommandos im Beispiel sehen wollen, sehen Sie bitte in das Beispielverzeichnis unter

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

Die Beispieldateien sind mit der benutzten Option bzw. dem benutzten Kommando benannt.

Eine Vorbemerkung zum Setzen von ersetztem Text: ersetzter Text wird immer wie folgt gesetzt: $\langle \text{neuer Text} \rangle \langle \text{alter Text} \rangle$. Daher gibt es keine Möglichkeit, die Ausgabe ersetzten Texts direkt zu beeinflussen, sondern nur über die Änderung der Ausgabe neuen bzw. gelöschten Texts.

4.1 Paketoptionen

4.1.1 draft

Die `draft`-Option bewirkt, dass alle Änderungen markiert werden. Die Änderungsliste kann durch `\listofchanges` ausgegeben werden. Diese Option ist automatisch voreingestellt.

Die Angabe von `draft` in `\documentclass` wird vom `changes`-Paket mitgenutzt. Die lokale Angabe von `final` überstimmt die Angabe von `draft` in `\documentclass`.

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

4.1.2 final

Die `final`-Option bewirkt, dass alle Änderungsmarkierungen ausgeblendet werden und nur noch der korrekte Text ausgegeben wird. Die Änderungsliste wird ebenfalls unterdrückt.

Die Angabe von `final` in `\documentclass` wird vom `changes`-Paket mitgenutzt. Die lokale Angabe von `draft` überstimmt die Angabe von `final` in `\documentclass`.

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

4.1.3 markup

Die `markup`-Option wählt ein vordefiniertes visuelles Markup für geänderten Text. Das `default`-Markup wird gewählt, wenn die Option nicht gesetzt wird. Das mit `markup` gewählte Markup kann mit den spezielleren Optionen `addedmarkup` und/oder `deletedmarkup` geändert werden.

Die folgenden Werte sind erlaubt:

default farbige Markierung von zugefügtem Text, gelöschter Text wird durchgestrichen (default-Markup)

underlined zugefügter Text wird unterstrichen, gelöschter Text wird durchgestrichen

bfit fatter zugefügter Text, schräger gelöschter Text

nocolor es werden keine Farben verwendet, zugefügter Text wird unterstrichen, gelöschter Text wird durchgestrichen

Aufruf

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

Beispiele

```
\usepackage[markup=default]{changes} ≅ \usepackage{changes}
\usepackage[markup=underlined]{changes}
\usepackage[markup=bfit]{changes}
\usepackage[markup=nocolor]{changes}
```

4.1.4 addedmarkup, deletedmarkup

Die `addedmarkup`-Option wählt ein vordefiniertes visuelles Markup für zugefügten Text. Die `deletedmarkup`-Option wählt analog ein vordefiniertes visuelles Markup für gelöschten Text. Das `default`-Markup wird gewählt, wenn die Option nicht gesetzt wird. Die Optionen `addedmarkup` und `deletedmarkup` überschreiben das mit `markup` gewählte Markup.

Die folgenden Werte sind erlaubt:

none kein Markup – Beispiel (default-Markup für zugefügten Text)

uline unterstrichener Text – *Beispiel*

uuline doppelt unterstrichener Text – *Beispiel*

uwave gewellt unterstrichener Text – *Beispiel*

dashuline gestrichelt unterstrichener Text – *Beispiel*

dotuline gepunktet unterstrichener Text – *Beispiel*

sout durchgestrichener Text – *Beispiel* (default-Markup für gelöschten Text)

xout schräg durchgestrichener Text – *Beispiel*

bf fatter Text – *Beispiel*

it italic Text – *Beispiel*

sl schräger Text – *Beispiel*

em hervorgehobener Text – *Beispiel*

Aufruf

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

Beispiele

```
\usepackage[addedmarkup=none]{changes} ≅ \usepackage{changes}
\usepackage[addedmarkup=uline]{changes}
```

Aufruf

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

Beispiele

```
\usepackage[deletedmarkup=sout]{changes} ≅ \usepackage{changes}
\usepackage[deletedmarkup=xout]{changes}
\usepackage[deletedmarkup=uwave]{changes}
```

4.1.5 authormarkup

Die authormarkup-Option wählt ein vordefiniertes visuelles Markup für die Autor-Identifizierung. Das default-Markup wird gewählt, wenn die Option nicht gesetzt wird.

Die folgenden Werte sind erlaubt:

superscript hochgestellter Text – Text^{Beispiel} (default-Markup)

subscript tiefgestellter Text – Text_{Beispiel}

brackets Text in Klammern – Text(Beispiel)

footnote Text in einer Fußnote – Text¹

Aufruf

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

Beispiele

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

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

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

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

4.1.6 authormarkupposition

Die authormarkupposition-Option gibt an, wo die Autor-Identifizierung gesetzt wird. Der default-Wert wird gewählt, wenn die Option nicht gesetzt wird.

Die folgenden Werte sind erlaubt:

right rechts vom Text – Text^{Beispiel} (default value)

left links vom Text – ^{Beispiel}Text

Aufruf

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

Beispiele

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

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

¹Beispiel

4.1.7 authormarkuptext

Die `authormarkuptext`-Option gibt an, was für die Autor-Identifizierung genutzt wird. Der default-Wert wird gewählt, wenn die Option nicht gesetzt wird.

Die folgenden Werte sind erlaubt:

id Autoren-ID – Text^{ID} (default-Wert)

name Autorennamen – Text^{Autorennamen}

Aufruf

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

Beispiele

```
\usepackage[authormarkuptext=id]{changes} ≅ \usepackage{changes}
```

```
\usepackage[authormarkuptext=name]{changes}
```

4.1.8 ulem

Optionen für das *ulem*-Paket können als Parameter der `ulem`-Option angegeben werden. Zwei oder mehr Optionen müssen in geschweifte Klammern gesetzt werden.

Aufruf

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

Beispiele

```
\usepackage[ulem=normalem]{changes}
```

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

4.1.9 xcolor

Optionen für das *xcolor*-Paket können als Parameter der `xcolor`-Option angegeben werden. Zwei oder mehr Optionen müssen in geschweifte Klammern gesetzt werden.

Aufruf

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

Beispiele

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

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

4.2 Änderungsmanagement

4.2.1 \added

\added

Der Befehl `\added` markiert zugefügten Text. Der neue Text wird als notwendiges Argument in geschweiften Klammern übergeben. Optional können eine Autor-ID sowie eine Anmerkung übergeben werden. Die Autor-ID muss mit einer mit dem `\definechangesauthor`-Befehl definierten ID übereinstimmen. Soll nur eine Anmerkung (ohne Autor) eingegeben werden, so ist statt des Autors ein leeres Argument zu übergeben.

Aufruf

```
\added[⟨Autor-ID⟩][⟨Anmerkung⟩]{⟨neuer Text⟩}
```

Beispiele

Das ist `\added[EK]{neuer}` Text.

Das ist neuer^{EK} Text.

Das ist `\added[EK][muss rein]{neuer}` Text.

Das ist neuer^{EK}(muss rein) Text.

Das ist `\added[][anonym]{neuer}` Text.

Das ist neuer(anonym) Text.

4.2.2 \deleted

\deleted

Der Befehl `\deleted` markiert gelöschten Text. Argumente: siehe `\added`.

Aufruf

```
\deleted[⟨Autor-ID⟩][⟨Anmerkung⟩]{⟨gelöschter Text⟩}
```

Beispiele

Das ist `\deleted[][obsolet]{schlechter}` Text.

Das ist schlechter(obsolet) Text.

4.2.3 \replaced

\replaced

Der Befehl `\replaced` markiert geänderten Text. Notwendige Argumente sind der neue sowie der alte Text. Optionale Argumente: siehe `\added`.

Aufruf

```
\replaced[<Autor-ID>][<Anmerkung>]{<neuer Text>}{<alter Text>}
```

Beispiele

Das ist \replaced[EK]{schöner}{schlechter} Text.

Das ist schöner^{EK}schlechter Text.

4.2.4 \listofchanges

`\listofchanges`

Der Befehl `\listofchanges` gibt eine Liste der Änderungen aus. Im ersten \TeX -Lauf wird eine Hilfsdatei angelegt, deren Daten im zweiten Durchlauf eingebunden werden. Für eine aktuelle Liste der Änderungen sind daher zwei \TeX -Läufe notwendig.

Aufruf

```
\listofchanges
```

4.3 Autorenverwaltung

4.3.1 \definechangesauthor

`\definechangesauthor`

Der Befehl `\definechangesauthor` definiert einen neuen Autor/eine neue Autorin für Änderungen. Es muss eine eindeutige Autor-ID angegeben werden, die keine Sonder- oder Leerzeichen enthalten darf. Optional kann eine Farbe und ein Name angegeben werden. Wird keine Farbe angegeben, wird schwarz genutzt. Der AutorInnenname wird in der Änderungsliste sowie im Markup benutzt, im Markup jedoch nur, wenn die entsprechende Option gesetzt ist.

Aufruf

```
\definechangesauthor[name={<author's name>}, color={<color>}]{<author's id>}
```

Beispiele

```
\definechangesauthor{EK}
```

```
\definechangesauthor[color=orange]{EK}
```

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

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

4.4 Anpassung der Ausgabe

4.4.1 `\setaddedmarkup`

`\setaddedmarkup`

Der Befehl `\setaddedmarkup` legt fest, wie neuer Text ausgezeichnet wird. Ohne andere Definition gilt, dass der Text farbig oder je nach Option `markup` bzw. `addedmarkup` erscheint.

Werte für Definition: beliebige \TeX -Befehle, der neue Text wird mit "`#1`" eingesetzt.

Aufruf

```
\setaddedmarkup{<definition>}
```

Beispiele

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

4.4.2 `\setdeletedmarkup`

`\setdeletedmarkup`

Der Befehl `\setdeletedmarkup` legt fest, wie gelöschter Text ausgezeichnet wird. Ohne andere Definition gilt, dass der Text durchgestrichen oder je nach Option `markup` bzw. `deletedmarkup` erscheint.

Werte für Definition: beliebige \TeX -Befehle, der gelöschte Text wird mit "`#1`" eingesetzt.

Aufruf

```
\setdeletedmarkup{<definition>}
```

Beispiele

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

4.4.3 `\setauthormarkup`

`\setauthormarkup`

Der Befehl `\setauthormarkup` legt fest, wie der Autortext im Text angezeigt wird. Ohne andere Definition gilt, dass der Autor hochgestellt erscheint.

Werte für Definition: beliebige \TeX -Befehle, der Autortext wird mit "`#1`" eingesetzt.

Aufruf

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

Beispiele

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

4.4.4 \setauthormarkupposition

`\setauthormarkupposition`

Der Befehl `\setauthormarkupposition` legt fest, auf welcher Seite der Autor im Text angezeigt wird. Ohne andere Definition gilt, dass der Autor rechts von den Änderungen erscheint.

Mögliche Werte: *left* == links von den Änderungen; alles andere: rechts

Aufruf

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

Beispiele

```
\setauthormarkupposition{left}
```

4.4.5 \setauthormarkuptext

`\setauthormarkuptext`

Der Befehl `\setauthormarkuptext` legt fest, welche Information des Autors im Text angezeigt wird. Ohne andere Definition gilt, dass die Autor-ID genutzt wird.

Mögliche Werte: *name* == Autorennamen; alles andere: Autor-ID

Aufruf

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

Beispiele

```
\setauthormarkuptext{name}
```


4.4.6 `\setremarkmarkup`

`\setremarkmarkup`

Der Befehl `\setremarkmarkup` legt fest, wie die Anmerkungen im Text angezeigt werden. Ohne andere Definition gilt, dass die Anmerkungen als Fußnote gesetzt werden.

Werte für Definition: beliebige \TeX -Befehle, die Autor-ID wird mit "`#1`" benutzt, der Anmerkungstext mit "`#2`". Über die Autor-ID kann mit `Changes@Color#1` die Farbe des Autors benutzt werden.

Aufruf

```
\setremarkmarkup{<definition>}
```

Beispiele

```
\setremarkmarkup{#2 -- #1}
```

```
\setremarkmarkup{\footnote{#1:\textcolor{Changes@Color#1}{#2}}}
```

4.4.7 `\setlocextension`

`\setlocextension`

Der Befehl `\setlocextension` legt das Suffix der Hilfsdatei für die Änderungsliste (loc-Datei²) fest. Ohne andere Definition gilt das Suffix "`loc`". Im unten angegebenen Beispiel würde für "`foo.tex`" eine Hilfsdatei erzeugt werden, die "`foo.changes`" statt des Standardnamens "`foo.loc`" hieße.

Aufruf

```
\setlocextension{<extension>}
```

Beispiele

```
\setlocextension{changes}
```

4.5 Sonstige neue Befehle

4.5.1 `\textsubscript`

`\textsubscript`

\TeX stellt den Befehl `\textsuperscript` zur Verfügung, nicht jedoch dessen Gegenstück `\textsubscript`. Falls der Befehl nicht bereits definiert ist, wird er durch das *changes*-Paket zur Verfügung gestellt. Ist er bereits definiert, wird er nicht geändert.

² "`loc`" steht dabei für "`list of changes`".

Aufruf

`\textsubscript{<Text>}`

Beispiele

Jetzt kommt ein `\textsubscript{tiefgestellter}` Text.

Jetzt kommt ein _{tiefgestellter} Text.

4.6 Benötigte Pakete

Das *changes*-Paket bindet bereits Pakete ein, die für die Funktion des Pakets notwendig sind. Eine genauere Beschreibung der einzelnen Pakete ist in der Dokumentation der Pakete selbst zu finden.

Die folgenden Pakete sind zwingend notwendig und müssen für die Nutzung des *changes*-Pakets installiert sein:

xifthen stellt eine verbesserte *if*-Abfrage sowie eine *while*-Schleife zur Verfügung
xkeyval Eingabe von Optionen mit Werteübergabe

Die folgenden Pakete sind manchmal notwendig und müssen installiert sein, wenn sie über die entsprechende Option genutzt werden:

pdfcolmk wird geladen, wenn farbiger Text genutzt wird (default Markup); löst das Problem farbigen Texts über Seitenumbrüche hinweg (bei *pdflatex*)

ulem wird geladen, wenn Text durchgestrichen oder *ausge-x-t* wird (default Markup)

xcolor wird geladen, wenn farbiger Text genutzt wird (default Markup)

5 Autoren

Am *changes*-Paket haben mehrere Autoren mitgewirkt. Dies sind in alphabetischer Reihenfolge:

- Chiaradonna, Silvano
- Giovannini, Daniele
- Kleinod, Ekkart
- Wölfel, Philipp
- Wolter, Steve

6 Versionen

Version 0.6.0

Datum: ?? .0?. 2012

- Italienische Übersetzungen der captions von Daniele Giovannini
- neues Nutzerinterface für das Setzen von Optionen sowie die Definition von Markup und Autoren
- Restrukturierung und Codeverbesserung
- verbesserte Dokumentation mit typischem Anwendungsfall
- Beispieldateien für alle Optionen und Befehle
- Anmerkungen sind per Default nicht mehr farbig

Version 0.5.4

Datum: 25.04. 2011

- Auslagerung der Nutzerdokumentation in eigene Datei
- Änderung der default-Sprache zu Englisch
- neues Script, um die *changes*-Befehle zu löschen von Silvano Chiaradonna

Version 0.5.3

Datum: 22.11. 2010

- Dokumentoptionen von `\documentclass` werden ebenfalls genutzt (Vorschlag und Code von Steve Wolter)

Version 0.5.2

Datum: 10.10. 2007

- Paketoptionen der Pakete *ulem* und *xcolor* werden weitergeleitet

Version 0.5.1

Datum: 27.08. 2007

- gelöschter Text wieder durchgestrichen, Paket *ulem* funktioniert; ausgrauen hat nicht funktioniert

Version 0.5

Datum: 26.08. 2007

- keine Nutzung des *arrayjob*-Pakets mehr, dadurch Fehler im Zusammenspiel mit *array* behoben

- auf UTF-8-encoding umgestellt
- keine Nutzung des *soul*-Pakets mehr, dadurch Fehler im Zusammenspiel UTF-8-encoding behoben
- gelöschter Text durch grauen Hintergrund visualisiert (es gibt bisher kein ordentliches Durchstreichen bei UTF-8-Nutzung)
- neues optionales Argument für Autorennamen
- farbige Liste der Änderungen
- loc-Format geändert
- englische Doku verbessert

Version 0.4

Datum: 24. 01. 2007

- *pdfcolmk* eingebunden, um Problem mit farbigem Text bei Seitenumbrüchen zu lösen
- `\setremarkmarkup` um Autor-ID erweitert, um Anmerkung farbig setzen zu können
- Anmerkungen werden in der Fußnote farbig gesetzt
- erste Version für das CTAN

Version 0.3

Datum: 22. 01. 2007

- englische Nutzerdokumentation
- Befehl `\changed` ersetzt durch `\replaced`
- verbesserte *final*-Option: kein zusätzlicher Leerraum

Version 0.2

Datum: 17. 01. 2007

- Bezeichnungen auch bei fehlendem *babel*-Paket eingeführt
- `\setauthormarkup`, `\setlocextension`, `\setremarkmarkup` für Einstellungen
- Beispieldateien generiert
- LPPL eingefügt

Beseitigte Fehler

- Fehler mit *ifthen*-Paketplatzierung behoben
- bei Liste war immer "Eingefügt" eingestellt, behoben
- Autorausgabe war buggy (`\if`-Abfrage nicht einwandfrei)

Version 0.1

Datum: 16. 01. 2007

- initiale Version
- Befehle `\added`, `\deleted` und `\changed`

7 Weitergabe, Copyright, Lizenz

Copyright 2007-2012 Ekkart Kleinod (ekleinod@edgesoft.de)

Dieses Paket darf unter der "L^AT_EX Project Public License" Version 1.3 oder jeder späteren Version weitergegeben und/oder geändert werden. Die neueste Version dieser Lizenz steht auf <http://www.latex-project.org/lppl.txt> Version 1.3 und spätere Versionen sind Teil aller L^AT_EX-Distributionen ab Version 2005/12/01.

Dieses Paket besitzt den Status "maintained" (verwaltet). Der aktuelle Verwalter dieses Pakets ist Ekkart Kleinod.

Dieses Paket besteht aus den Dateien

```
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
```

und den generierten Dateien

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

8 The documented sourcecode

The sourcecode is documented in English only. This is intended, please do not provide translations for the text below, just corrections or improvements.

```
1 ⟨*changes⟩
```

8.1 Package information and options

Set needed \LaTeX -format to $\text{\LaTeX}2_{\epsilon}$, provide name, date, version. Type some information to the console.

```
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{changes}
4 [2011/10/19 v0.6.0 changes-Paket]
5 \typeout{*** changes-Paket 2011/10/19 v0.6.0 ***}
```

Package *xkeyval* provides options with key-value-pairs.

```
6 \RequirePackage{xkeyval}
```

Package *xifthen* provides improved if as well as a while-loop.

```
7 \RequirePackage{xifthen}
```

8.1.1 Package options

Option *draft*, *default* is true.

```
8 \newboolean{Changes@optiondraft}
9 \setboolean{Changes@optiondraft}{true}
10 \DeclareOptionX{draft}{
11 \setboolean{Changes@optiondraft}{true}
12 \typeout{changes-option '\CurrentOption'}
13 }
```

Option *final*, sets *draft* to false.

```
14 \DeclareOptionX{final}{
15 \setboolean{Changes@optiondraft}{false}
16 \typeout{changes-option '\CurrentOption'}
17 }
```

Declare storage for markup options, they are set by the markup option but can be changed with the more special options, therefore they have to be declared at this place.

```
18 \newcommand{\Changes@optionaddedmarkup}{none}
19 \newcommand{\Changes@optiondeletedmarkup}{sout}
```

Option markup, sets markup options accordingly.

```
20 \newcommand{\Changes@optionmarkup}{default}
21 \DeclareOptionX{markup}{
22 \ifthenelse{\equal{\@empty}{#1}}
23 {}
24 {
25 \ifthenelse{
26 \equal{#1}{default}\or
27 \equal{#1}{underlined}\or
28 \equal{#1}{bfit}\or
29 \equal{#1}{nocolor}
30 }
31 {\renewcommand{\Changes@optionmarkup}{#1}}
32 {\PackageWarning{changes}{markup '#1' unknown, using '\Changes@optionmarkup'}}
33 }
34 \ifthenelse{\equal{\Changes@optionmarkup}{default}}
35 {
36 \renewcommand{\Changes@optionaddedmarkup}{none}
37 \renewcommand{\Changes@optiondeletedmarkup}{sout}
38 }
39 {}
40 \ifthenelse{\equal{\Changes@optionmarkup}{underlined}}
41 {
42 \renewcommand{\Changes@optionaddedmarkup}{uline}
43 \renewcommand{\Changes@optiondeletedmarkup}{sout}
44 }
45 {}
46 \ifthenelse{\equal{\Changes@optionmarkup}{bfit}}
47 {
48 \renewcommand{\Changes@optionaddedmarkup}{bf}
49 \renewcommand{\Changes@optiondeletedmarkup}{it}
50 }
51 {}
52 \ifthenelse{\equal{\Changes@optionmarkup}{nocolor}}
53 {
54 \renewcommand{\Changes@optionaddedmarkup}{uline}
55 \renewcommand{\Changes@optiondeletedmarkup}{sout}
56 }
57 {}
58 \typeout{changes-option 'markup'=\Changes@optionmarkup}
59 }
```

Option addedmarkup, stored or set to default value “none”.

```
60 \DeclareOptionX{addedmarkup}{
61 \ifthenelse{\equal{\@empty}{#1}}
62 {}
63 {
64 \ifthenelse{
```

```
65 \equal{#1}{none}\or
66 \equal{#1}{uline}\or
67 \equal{#1}{uuline}\or
68 \equal{#1}{uwave}\or
69 \equal{#1}{dashuline}\or
70 \equal{#1}{dotuline}\or
71 \equal{#1}{sout}\or
72 \equal{#1}{xout}\or
73 \equal{#1}{bf}\or
74 \equal{#1}{it}\or
75 \equal{#1}{sl}\or
76 \equal{#1}{em}
77 }
78 {\renewcommand{\Changes@optionaddedmarkup}{#1}}
79 {\PackageWarning{changes}{addedmarkup '#1' unknown, using '\Changes@optionaddedmarkup'}}
80 }
81 \typeout{changes-option 'addedmarkup=\Changes@optionaddedmarkup'}
82 }
```

Option `deletedmarkup`, stored or set to default value “`striked`”.

```
83 \DeclareOptionX{deletedmarkup}{
84 \ifthenelse{\equal{\@empty}{#1}}
85 {}
86 {
87 \ifthenelse{
88 \equal{#1}{none}\or
89 \equal{#1}{uline}\or
90 \equal{#1}{uuline}\or
91 \equal{#1}{uwave}\or
92 \equal{#1}{dashuline}\or
93 \equal{#1}{dotuline}\or
94 \equal{#1}{sout}\or
95 \equal{#1}{xout}\or
96 \equal{#1}{bf}\or
97 \equal{#1}{it}\or
98 \equal{#1}{sl}\or
99 \equal{#1}{em}
100 }
101 {\renewcommand{\Changes@optiondeletedmarkup}{#1}}
102 {\PackageWarning{changes}{deletedmarkup '#1' unknown, using '\Changes@optiondeletedmarkup'}}
103 }
104 \typeout{changes-option 'deletedmarkup=\Changes@optiondeletedmarkup'}
105 }
```

Declare storage for `authormarkup` option and store option value or set to default value “`superscript`”.

```
106 \newcommand{\Changes@optionauthormarkup}{superscript}
107 \DeclareOptionX{authormarkup}{
```



```
108 \ifthenelse{\equal{\@empty}{#1}}
109 {}
110 {
111 \ifthenelse{
112 \equal{#1}{superscript}\or
113 \equal{#1}{subscript}\or
114 \equal{#1}{brackets}\or
115 \equal{#1}{footnote}
116 }
117 {\renewcommand{\Changes@optionauthormarkup}{#1}}
118 {\PackageWarning{changes}{authormarkup '#1' unknown, using '\Changes@optionauthormarkup'}}
119 }
120 \typeout{changes-option 'authormarkup'=\Changes@optionauthormarkup'}
121 }
```

Declare storage for authormarkupposition option and store option value or set to default value “right”.

```
122 \newcommand{\Changes@optionauthormarkupposition}{right}
123 \DeclareOptionX{authormarkupposition}{
124 \ifthenelse{\equal{\@empty}{#1}}
125 {}
126 {
127 \ifthenelse{
128 \equal{#1}{right}\or
129 \equal{#1}{left}
130 }
131 {\renewcommand{\Changes@optionauthormarkupposition}{#1}}
132 {\PackageWarning{changes}{authormarkupposition '#1' unknown, using '\Changes@optionauthormarkupposition'}}
133 }
134 \typeout{changes-option 'authormarkupposition'=\Changes@optionauthormarkupposition'}
135 }
```

Declare storage for authormarkuptext option and store option value or set to default value “id”.

```
136 \newcommand{\Changes@optionauthormarkuptext}{id}
137 \DeclareOptionX{authormarkuptext}{
138 \ifthenelse{\equal{\@empty}{#1}}
139 {}
140 {
141 \ifthenelse{
142 \equal{#1}{id}\or
143 \equal{#1}{name}
144 }
145 {\renewcommand{\Changes@optionauthormarkuptext}{#1}}
146 {\PackageWarning{changes}{authormarkuptext '#1' unknown, using '\Changes@optionauthormarkuptext'}}
147 }
148 \typeout{changes-option 'authormarkuptext'=\Changes@optionauthormarkuptext'}
149 }
```

Options for package *ulem* are directly passed to the package.

```
150 \DeclareOptionX{ulem}{  
151 \typeout{ulem-option '#1', passed to package ulem}  
152 \PassOptionsToPackage{#1}{ulem}  
153 }
```

Options for package *xcolor* are directly passed to the package.

```
154 \DeclareOptionX{xcolor}{  
155 \typeout{xcolor-option '#1', passed to package xcolor}  
156 \PassOptionsToPackage{#1}{xcolor}  
157 }
```

Unknown options generate a package warning.

```
158 \DeclareOptionX*{  
159 \PackageWarning{changes}{Unknown option '\CurrentOption'}  
160 }
```

8.1.2 Command options

All options for commands (e.g. `\definechangesauthor`) have to be declared before option processing.

Declare available options of the command, define value storage.

```
161 \DeclareOptionX<Changes@definechangesauthor>{name}{\def\Changes@definechangesauthor@name{#1}  
162 \DeclareOptionX<Changes@definechangesauthor>{color}{\def\Changes@definechangesauthor@color{
```

Set the default values of the options.

```
163 \presetkeys{Changes@definechangesauthor}{  
164 name=\@empty,  
165 color=black  
166 }{}
```

8.1.3 Option processing

Process the options.

```
167 \ProcessOptionsX*\relax
```

8.2 Packages

Package *xcolor* provides colored text. Package *pdfcolmk* solves the problem of colored text and page breaks (has to be loaded after *xcolor*).

```

168 \newboolean{Changes@colored}
169 \setboolean{Changes@colored}{true}
170 \ifthenelse{\equal{\Changes@optionmarkup}{nocolor}}
171 {\setboolean{Changes@colored}{false}}
172 {}
173 \ifthenelse{\boolean{Changes@colored}}
174 {
175   \RequirePackage{xcolor}
176   \RequirePackage{pdfcolmk}
177 }
178 {}

```

Package *ulem* provides commands for striking out text.

```

179 \ifthenelse{
180 \equal{\Changes@optionaddedmarkup}{uline}\or
181 \equal{\Changes@optionaddedmarkup}{uuline}\or
182 \equal{\Changes@optionaddedmarkup}{uwave}\or
183 \equal{\Changes@optionaddedmarkup}{dashuline}\or
184 \equal{\Changes@optionaddedmarkup}{dotuline}\or
185 \equal{\Changes@optionaddedmarkup}{sout}\or
186 \equal{\Changes@optionaddedmarkup}{xout}\or
187 \equal{\Changes@optiondeletedmarkup}{uline}\or
188 \equal{\Changes@optiondeletedmarkup}{uuline}\or
189 \equal{\Changes@optiondeletedmarkup}{uwave}\or
190 \equal{\Changes@optiondeletedmarkup}{dashuline}\or
191 \equal{\Changes@optiondeletedmarkup}{dotuline}\or
192 \equal{\Changes@optiondeletedmarkup}{sout}\or
193 \equal{\Changes@optiondeletedmarkup}{xout}
194 }
195 {\RequirePackage[normalem,normalbf]{ulem}}
196 {}

```

8.3 Language dependent texts

Declaration of language dependent names and identifiers. The check for `\addto` is a check for the *babel* package. If the *babel* package is not loaded, the default language is English, in order to use another language, the user has to redefine the variables.

```

197 \ifthenelse{\isundefined{\addto}}
198 {
199 \def\listchangesname{Changes}

```

```
200 \def\changesaddname{Added}
201 \def\changesdeletename{Deleted}
202 \def\changesreplacename{Replaced}
203 \def\changesauthorname{Author}
204 \def\changesanonymousname{anonymous}
205 \def\changesnoloc{List of changes is available after the next \LaTeX\ run.}
206 }{
207 \addto\captionsngerman{\def\listchangesname{"Anderungen}}
208 \addto\captionsngerman{\def\changesaddname{Eingef\ "ugt}}
209 \addto\captionsngerman{\def\changesdeletename{Gel\ "oscht}}
210 \addto\captionsngerman{\def\changesreplacename{Ersetzt}}
211 \addto\captionsngerman{\def\changesauthorname{Autor}}
212 \addto\captionsngerman{\def\changesanonymousname{Anonym}}
213 \addto\captionsngerman{\def\changesnoloc{"Anderungsliste nach dem n\ "achsten \LaTeX-Lauf v
214
215 \addto\captionsngerman{\def\listchangesname{"Anderungen}}
216 \addto\captionsngerman{\def\changesaddname{Eingef\ "ugt}}
217 \addto\captionsngerman{\def\changesdeletename{Gel\ "oscht}}
218 \addto\captionsngerman{\def\changesreplacename{Ersetzt}}
219 \addto\captionsngerman{\def\changesauthorname{Autor}}
220 \addto\captionsngerman{\def\changesanonymousname{Anonym}}
221 \addto\captionsngerman{\def\changesnoloc{"Anderungsliste nach dem n\ "achsten \LaTeX-Lauf v
222
223 \addto\captionsenglish{\def\listchangesname{Changes}}
224 \addto\captionsenglish{\def\changesaddname{Added}}
225 \addto\captionsenglish{\def\changesdeletename{Deleted}}
226 \addto\captionsenglish{\def\changesreplacename{Replaced}}
227 \addto\captionsenglish{\def\changesauthorname{Author}}
228 \addto\captionsenglish{\def\changesanonymousname{anonymous}}
229 \addto\captionsenglish{\def\changesnoloc{List of changes is available after the next \LaTeX
230
231 \addto\captionsitalian{\def\listchangesname{Modifiche}}
232 \addto\captionsitalian{\def\changesaddname{Aggiunte}}
233 \addto\captionsitalian{\def\changesdeletename{Cancellazioni}}
234 \addto\captionsitalian{\def\changesreplacename{Sostituzioni}}
235 \addto\captionsitalian{\def\changesauthorname{Autore}}
236 \addto\captionsitalian{\def\changesanonymousname{anonimo}}
237 \addto\captionsitalian{\def\changesnoloc{La lista delle modifiche sar\ 'a disponibile alla p
238 }
```

8.4 File extension

`\Changes@extension` Store file extension in variable, set default to loc.

```
239 \newcommand{\Changes@extension}{loc}
```

`\setlocextension` Set a new file extension. Argument: new extension.

```

240 \newcommand{\setlocextension}[1]{
241 \renewcommand{\Changes@extension}{#1}
242 }

```

8.5 Authors

8.5.1 Author management

Author counter.

```

243 \newcounter{Changes@AuthorCount}
244 \setcounter{Changes@AuthorCount}{0}
245 \newcounter{Changes@iAuthor}

```

`\definechangesauthor` Define a new author. Mandatory argument: author's id. Optional arguments (key-value): author's name (default: empty) and author's color (default: black).

Store id, name and color using named variables. Define counter and color per author.

```

246 \newcommand*\definechangesauthor[2][\]{%

```

Call `setkeys` in order to evaluate the key-value-options and fill the value storage.

```

247 \setkeys{Changes@definechangesauthor}{#1}

```

Increment author counter, later needed for *while* loop of authors.

```

248 \stepcounter{Changes@AuthorCount}

```

Store the id in a name with the given counter/index. All other storage refers to the id.

```

249 \@namedef{Changes@AuthorID\theChanges@AuthorCount}{#2}

```

Store the author's definition in according variables/colors, create change counters.

```

250 \expandafter
251 \let\csname Changes@AuthorName#2\endcsname=\Changes@definechangesauthor@name
252 \newcounter{Changes@AddCount#2}
253 \newcounter{Changes@DeleteCount#2}
254 \newcounter{Changes@ReplaceCount#2}
255 \ifthenelse{\boolean{Changes@colored}}{
256 {
257 \expandafter
258 \let\csname Changes@AuthorColor#2\endcsname=\Changes@definechangesauthor@color
259 \colorlet{Changes@Color#2}{\@nameuse{Changes@AuthorColor#2}}
260 }

```

```
261 {}  
262 }
```

Define default-author (anonymous) with empty id and blue color.

```
263 \definechangesauthor[color=blue]{\@empty}
```

8.5.2 Author markup

`\changes@Markup@Author` Store markup for authors.

```
264 \newcommand{\Changes@Markup@Author}[1]{%  
265 \ifthenelse{\equal{\Changes@optionauthormarkup}{superscript}}{\textsuperscript{#1}}{}%  
266 \ifthenelse{\equal{\Changes@optionauthormarkup}{subscript}}{\textsubscript{#1}}{}%  
267 \ifthenelse{\equal{\Changes@optionauthormarkup}{brackets}}{(#1)}{}%  
268 \ifthenelse{\equal{\Changes@optionauthormarkup}{footnote}}{\footnote{#1}}{}%  
269 }
```

`\setauthormarkup` Set markup for authors.

```
270 \newcommand{\setauthormarkup}[1]{  
271 \renewcommand{\Changes@Markup@Author}[1]{#1}  
272 }
```

`\textsubscript` Define the command `\textsubscript` in case the author markup subscript is used and the command `\textsubscript` is not defined yet. `\textsubscript` is the antagonist of `\textsuperscript` which is predefined in \LaTeX . The code is taken from \LaTeX -FAQ 8.5.17.

```
273 \ifthenelse{\isundefined{\textsubscript}}  
274 {  
275 \DeclareRobustCommand*\textsubscript[1]{\@textsubscript{\selectfont#1}}  
276 \newcommand{\@textsubscript}[1]{\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}  
277 }
```

`\authormarkupposition` Set position for author markup text.

```
278 \newcommand{\setauthormarkupposition}[1]{  
279 \renewcommand{\Changes@optionauthormarkupposition}{#1}  
280 }
```

`\setauthormarkuptext` Set author markup text to be displayed.

```
281 \newcommand{\setauthormarkuptext}[1]{  
282 \renewcommand{\Changes@optionauthormarkuptext}{#1}  
283 }
```

`\Changes@Remark` Markup of remarks. Default: in a footnote.

```
284 \newcommand{\Changes@Remark}[2]{%
285 \footnote{%
286 \ifthenelse{\not\equal{#1}{\@empty}}%
287 {#1: }%
288 }%
289 #2%
290 }%
291 }
```

`\setremarkmarkup` Redefining the remark markup. Mandatory argument: markup definition.

```
292 \newcommand{\setremarkmarkup}[1]{%
293 \renewcommand{\Changes@Remark}[2]{#1}%
294 }
```

8.6 Change management commands

8.6.1 Text markup definition

Replaced text is always typeset as follows: `<added text><deleted text>`. Therefore no extra command for markup of replaced text is given.

`\Changes@Markup@Added` Store markup for added text.

```
295 \newcommand{\Changes@Markup@Added}[1]{%
296 \ifthenelse{\equal{\Changes@optionaddedmarkup}{none}}{#1}{}%
297 \ifthenelse{\equal{\Changes@optionaddedmarkup}{uline}}{\uline{#1}}{}%
298 \ifthenelse{\equal{\Changes@optionaddedmarkup}{uuline}}{\uuline{#1}}{}%
299 \ifthenelse{\equal{\Changes@optionaddedmarkup}{uwave}}{\uwave{#1}}{}%
300 \ifthenelse{\equal{\Changes@optionaddedmarkup}{dashuline}}{\dashuline{#1}}{}%
301 \ifthenelse{\equal{\Changes@optionaddedmarkup}{dotuline}}{\dotuline{#1}}{}%
302 \ifthenelse{\equal{\Changes@optionaddedmarkup}{sout}}{\sout{#1}}{}%
303 \ifthenelse{\equal{\Changes@optionaddedmarkup}{xout}}{\xout{#1}}{}%
304 \ifthenelse{\equal{\Changes@optionaddedmarkup}{bf}}{\textbf{#1}}{}%
305 \ifthenelse{\equal{\Changes@optionaddedmarkup}{it}}{\textit{#1}}{}%
306 \ifthenelse{\equal{\Changes@optionaddedmarkup}{sl}}{\textsl{#1}}{}%
307 \ifthenelse{\equal{\Changes@optionaddedmarkup}{em}}{\emph{#1}}{}%
308 }
```

`\setaddedmarkup` Set markup for added text.

```
309 \newcommand{\setaddedmarkup}[1]{
310 \renewcommand{\Changes@Markup@Added}[1]{#1}
311 }
```

`\changes@Markup@Deleted` Store markup for deleted text.

```

312 \newcommand{\Changes@Markup@Deleted}[1]{%
313 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{none}}{\#1}{}%
314 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{uline}}{\uline{\#1}}{}%
315 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{uuline}}{\uuline{\#1}}{}%
316 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{uwave}}{\uwave{\#1}}{}%
317 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{dashuline}}{\dashuline{\#1}}{}%
318 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{dotuline}}{\dotuline{\#1}}{}%
319 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{sout}}{\sout{\#1}}{}%
320 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{xout}}{\xout{\#1}}{}%
321 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{bf}}{\textbf{\#1}}{}%
322 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{it}}{\textit{\#1}}{}%
323 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{sl}}{\textsl{\#1}}{}%
324 \ifthenelse{\equal{\Changes@optiondeletedmarkup}{em}}{\emph{\#1}}{}%
325 }
```

`\setdeletedmarkup` Set markup for deleted text.

```

326 \newcommand{\setdeletedmarkup}[1]{
327 \renewcommand{\Changes@Markup@Deleted}[1]{\#1}
328 }
```

8.6.2 Change management command definition

`\Changes@output` Output command for the changed text. This command has the following arguments:

1. changed text (including markup)
2. unchanged text
3. author's id
4. remark

```
329 \newcommand{\Changes@output}[4]{%
```

Output changed text if option draft is set, otherwise output unchanged text.

```

330 \ifthenelse{\boolean{Changes@optiondraft}}{%
331 {%
```

Save author text for output.

```

332 \ifthenelse{\equal{\Changes@optionauthormarkuptext}{id}}{%
333 {\@namedef{Changes@AuthorText}{\#3}}%
334 }%
335 \ifthenelse{\equal{\Changes@optionauthormarkuptext}{name}}{%
336 {\@namedef{Changes@AuthorText}{\@nameuse{Changes@AuthorName\#3}}}%
337 }%
338 {%
```


Change color, if colored text is used.

```
339 \ifthenelse{\boolean{Changes@colored}}{%
340 {\color{Changes@Color#3}}%
341 {}}%
```

Output author text if author's id is given and text should appear left of changes.

```
342 \ifthenelse{\equal{\Changes@optionauthormarkupposition}{left} \and \not\equal{#3}{\@empty}}{%
343 {\Changes@Markup@Author{\@nameuse{Changes@AuthorText}}}%
344 {}}%
```

Output changed text.

```
345 {#1}%
```

Output author text if author's id is given and text should appear right of changes.

```
346 \ifthenelse{\equal{\Changes@optionauthormarkupposition}{right} \and \not\equal{#3}{\@empty}}{%
347 {\Changes@Markup@Author{\@nameuse{Changes@AuthorText}}}%
348 {}}%
```

Output remark if a remark is given.

```
349 \ifthenelse{\not\equal{#4}{\@empty}}{%
350 {\Changes@Remark{#3}{#4}}%
351 {}}%
352 }%
353 }%
```

Output unchanged text (option final was set).

```
354 {#2}%
355 }
```

`\Changes@temp` Temporary variable for exchange of optional parameters between interlocked commands.

```
356 \newcommand{\Changes@temp}{\@empty}
```

`\added` The command formats text as new text. Defining a command with two optional parameters is rather complicated.

Mandatory argument: added text. Optional arguments: author's id, remark

```
357 \newcommand{\added}[1][\@empty]{%
358 \renewcommand{\Changes@temp}{#1}%
359 \Changes@added%
360 }
361 \newcommand{\Changes@added}[2][\@empty]{%
```

```
362 \Changes@output
363 {\Changes@Markup@Added{#2}}
364 {#2}
365 {\Changes@temp}
366 {#1}%
367 \stepcounter{Changes@AddCount\Changes@temp}%
368 }
```

`\deleted` The command formats text as deleted text.

The definition of the empty text for unchanged text is taken from a tip from `de.comp.text.tex`. It solves the problem of additional space caused by an empty command.

Mandatory argument: deleted text. Optional arguments: author's id, remark

```
369 \newcommand{\deleted}[1][\@empty]{%
370 \renewcommand{\Changes@temp}{#1}%
371 \Changes@deleted%
372 }
373 \newcommand{\Changes@deleted}[2][\@empty]{%
374 \Changes@output
375 {\Changes@Markup@Deleted{#2}}
376 {\@bsphack \expandafter \@esphack}
377 {\Changes@temp}
378 {#1}%
379 \stepcounter{Changes@DeleteCount\Changes@temp}%
380 }
```

`\replaced` The command formats text as replaced text.

Mandatory arguments: new text and old text. Optional arguments: author's id, remark

```
381 \newcommand{\replaced}[1][\@empty]{%
382 \renewcommand{\Changes@temp}{#1}%
383 \Changes@replaced%
384 }
385 \newcommand{\Changes@replaced}[3][\@empty]{%
386 \Changes@output
387 {\{\Changes@Markup@Added{#2}}{\Changes@Markup@Deleted{#3}}}
388 {#2}
389 {\Changes@temp}
390 {#1}%
391 \stepcounter{Changes@ReplaceCount\Changes@temp}%
392 }
```

8.7 List of changes

`\changes@chopline` Auxiliary command for reading the content of the loc-files. The content is read line by line. One line is parsed with this macro, the order of entries is: id, color, name, added, deleted, replaced. The contents have to be separated by a semicolon.

```
393 \def\changes@chopline#1;#2;#3;#4;#5;#6 \{\
394 \def\Changes@InID{#1}
395 \def\Changes@InColor{#2}
396 \def\Changes@InName{#3}
397 \def\Changes@InAdded{#4}
398 \def\Changes@InDeleted{#5}
399 \def\Changes@InReplaced{#6}
400 }
```

`\listofchanges` This command outputs the list of changes, separated by authors. The values are read from the loc-file, if it exists. If no loc-file exists, an according message is generated.

```
401 \newcommand{\listofchanges}{%
402 \ifthenelse{\boolean{Changes@optiondraft}}{
403 {
404 \section*{\listchangesname}
405 \IfFileExists{\jobname.\Changes@extension}
406 {
407 \newboolean{Changes@MoreLines}
408 \setboolean{Changes@MoreLines}{true}
409 \newread\Changes@InFile
410 \openin\Changes@InFile = \jobname.\Changes@extension
411 \whiledo{\boolean{Changes@MoreLines}}{
412 \read\Changes@InFile to \Changes@Line
413 \ifeof\Changes@InFile
414 \setboolean{Changes@MoreLines}{false}
415 \else
416 \expandafter\changes@chopline\Changes@Line\
417 \begin{tabbing}
418 mm\=mmmmmm\=kill
419 {
420 \ifthenelse{\boolean{Changes@colored}}{
421 {\color{\Changes@InColor}}
422 {}
423 \ifthenelse{\equal{\Changes@InID}{\@empty}}{
424 {\changesauthorname: \changesanonymousname}
425 {
426 \changesauthorname: \Changes@InID
427 \ifthenelse{\equal{\Changes@InName}{\@empty}}{
428 {}
429 {(\Changes@InName)}
```

```
430 }
431 } \
432 \>\changesaddname:\>\Changes@InAdded\
433 \>\changesdeletename:\>\Changes@InDeleted\
434 \>\changesreplacename:\>\Changes@InReplaced\
435 \end{tabbing}
436 \fi
437 }
438 \closein\Changes@InFile
439 }{
440 \emph{\changesnoloc}
441 \PackageWarning{changes}{LaTeX rerun needed for list of changes}
442 }
443 }{}
444 }
```

At the end of the document: write the list of changes in the loc-file, therefore open file, write values, close file. Changes are written as \LaTeX -formatted text, so they can simply be read via `\input`.

The order of entries is: id, color, name, added, deleted, replaced. The contents have to be separated by a semicolon.

```
445 \AtEndDocument{
446 \newwrite\Changes@OutFile
447 \immediate\openout\Changes@OutFile = \jobname.\Changes@extension
448 \setcounter{Changes@iAuthor}{0}
449 \whiledo{\value{Changes@iAuthor} < \value{Changes@AuthorCount}}{
450 \stepcounter{Changes@iAuthor}
451 \def\Changes@ID{\@nameuse{Changes@AuthorID}\theChanges@iAuthor}}
452 \immediate\write\Changes@OutFile{\Changes@ID;%
453 \@nameuse{Changes@AuthorColor}\Changes@ID};%
454 \@nameuse{Changes@AuthorName}\Changes@ID};%
455 \the\value{Changes@AddCount}\Changes@ID};%
456 \the\value{Changes@DeleteCount}\Changes@ID};%
457 \the\value{Changes@ReplaceCount}\Changes@ID}}
458 }
459 \closeout\Changes@OutFile
460 }

461 </changes>
```