

Das *changes*-Paket

Manuelle Änderungsmarkierung – Version 4.0.2

5. April 2021

Ekkart Kleinod

@ ekleinod@edgesoft.de

1	Einleitung	4
2	Benutzung des <i>changes</i>-Pakets	5
3	Einschränkungen und Erweiterungsmöglichkeiten	8
4	Die Benutzerschnittstelle des <i>changes</i>-Pakets	9
4.1	Paketoptionen	9
4.1.1	draft	10
4.1.2	final	10
4.1.3	commandnameprefix	10
4.1.4	markup	11
4.1.5	addedmarkup	12
4.1.6	deletedmarkup	12
4.1.7	highlightmarkup	13
4.1.8	commentmarkup	14
4.1.9	authormarkup	14
4.1.10	authormarkupposition	15
4.1.11	authormarkuptext	15
4.1.12	defaultcolor	15
4.1.13	todonotes	16
4.1.14	truncate	16
4.1.15	ulem	16
4.1.16	xcolor	17
4.2	Änderungsmanagement	17
4.2.1	\added	17
4.2.2	\deleted	18
4.2.3	\replaced	18
4.3	Hervorhebungen und Kommentare	19
4.3.1	\highlight	19
4.3.2	\comment	19

4.4	Änderungsübersicht	20
4.4.1	<code>\listofchanges</code>	20
4.5	Autorenverwaltung	21
4.5.1	<code>\definechangesauthor</code>	21
4.6	Anpassung der Ausgabe	22
4.6.1	Werte für die Anpassung in den Makros	22
4.6.2	<code>\setaddedmarkup</code>	23
4.6.3	<code>\setdeletedmarkup</code>	23
4.6.4	<code>\sethighlightmarkup</code>	24
4.6.5	<code>\setcommentmarkup</code>	24
4.6.6	<code>\setauthormarkup</code>	25
4.6.7	<code>\setauthormarkupposition</code>	25
4.6.8	<code>\setauthormarkuptext</code>	25
4.6.9	<code>\settruncatewidth</code>	26
4.6.10	<code>\setsummarywidth</code>	26
4.6.11	<code>\setsummarytowidth</code>	26
4.6.12	<code>\setlocextension</code>	27
4.6.13	<code>\setsoextension</code>	27
4.7	Benötigte Pakete	28
5	Markierungen aus den Dateien entfernen	29
6	Bekannte Probleme und Lösungen	30
6.1	Besondere Inhalte	30
6.2	Fußnoten und Randnotizen	30
6.3	Das <i>ulem</i> -Paket	30
6.4	Kommando bereits definiert – command already defined	31
7	Autorinnen und Autoren	32
8	Versionen	33

1 Einleitung

Dieses Paket dient dazu, manuelle Änderungsmarkierung zu ermöglichen.

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

<https://edgesoft.de/projects/changes/>

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

Das changes-Paket dient zur manuellen Markierung von geändertem Text, insbesondere Einfügungen, Löschungen und Ersetzungen. Der geänderte Text wird farbig markiert und, bei gelöschtem Text, durchgestrichen. Zusätzlich kann Text hervorgehoben und/oder kommentiert werden. Das Paket ermöglicht die freie Definition von Autoren und deren zugeordneten Farben. Es erlaubt zusätzlich die Änderung des Änderungs-, Autor-, Hervorhebungs- und Kommentarmarkups.

Ein kurzes Beispiel für Änderungsmarkierung:

[EK 1] fehlendes Wort

Das ist zugefügter Text. In diesem Satz ersetze ich ein gutesschlechtes^{EK} Wort. Und jetzt noch ein schlechtes^{EK} Wort zum Löschen. Text kann auch hervorgehoben^{EK} oder nur kommentiert werden.

[EK 2] Aus Spaß!

Im gleichen Ordner wie dieses Handbuch befindet sich ein Ordner „examples“, der eine reichhaltige Auswahl an Anwendungsbeispielen für das Paket und dessen Befehle enthält. Bitte sehen Sie die Beispiele als Inspiration oder erste Fehlerlösungsquelle an.

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 Paketooptionen und neuen Befehle finden Sie nicht hier, sondern in Kapitel 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. Zusätzlich kann Text hervorgehoben oder kommentiert werden. Um das Paket zu nutzen, sollten Sie folgende Schritte ausführen:

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

***changes*-Paket einbinden**

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

```
\usepackage{changes}
```

bzw.

```
\usepackage[<options>]{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 Abschnitt 4.1 und Abschnitt 4.6.

Autoren definieren

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

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

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

Für Details lesen Sie bitte Abschnitt 4.5.

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

für gelöschten Text:

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

für geänderten Text:

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

Die Angabe von Autoren-ID und eines Kommentars ist optional.

Für Details lesen Sie bitte Abschnitt 4.2.

Text hervorheben und kommentieren

Vielleicht möchten Sie noch Text hervorheben oder kommentieren?

Text hervorheben:

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

Text kommentieren:

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

Die Angabe der Autoren-ID und des Kommentars für Hervorhebungen ist optional.

Für Details lesen Sie bitte Abschnitt 4.3.

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[style=<style>, title=<title>, show=<type>]
```

Die Ausgabe ist gedacht als Analogon zur Liste von Tabellen oder Abbildungen.

Die Angabe des Stils ist optional, standardmäßig wird `style=list` gewählt. Um einen schnellen Überblick über Art und Anzahl der Änderungen abhängig von dem/der Autor_in zu bekommen, verwenden Sie den Befehl mit der Option `style=summary` oder `style=compactsummary`. Zeigen Sie nur bestimmte Änderungstypen mit der `show`-Option.

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

Für Details lesen Sie bitte Abschnitt 4.4.

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}
```

Die Entfernung der Markierungen aus dem Quelltext müssen Sie von Hand vornehmen, dafür steht auch ein Script von Yvon Cui zur Verfügung. Das Script liegt im Verzeichnis:

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

Das Script entfernt alle Markierungen, indem die Änderungen angenommen oder abgelehnt werden. Sie können die zu entfernenden Markierungen individuell im interaktiven Modus selektieren bzw. selektieren, indem Sie das Skript ohne Optionen starten.

Für Details lesen Sie bitte Kapitel 5.

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.

Eine Übersicht über die wichtigsten mir bekannten Probleme und eventuell vorhandenen Lösungen finden Sie in Kapitel 6. Bitte sehen Sie dort zunächst nach, ob Ihr Problem schon bekannt ist und es eine Lösung gibt. Weitere Fehler, Probleme und Lösungen finden Sie auf:

<https://edgesoft.de/projects/changes/>

or

<https://gitlab.com/ekleinod/changes/-/issues>

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 ist eingeschränkt oder nicht möglich für:

- Abbildungen
- Tabellen
- Überschriften
- manche Kommandos
- mehrere Absätze (manchmal)

Sie können versuchen, solchen Text in eine eigene Datei auszulagern, und diese mit `input` einzubinden. Manchmal hilft das, oft ist es einen Versuch wert. Danke an Charly Arenz für diesen Tip.

Wenn Sie Fehler wegen bereits vorhandener Makros bekommen, schauen Sie sich bitte die Option `commandnameprefix` in Abschnitt 4.1.3 an.

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 Beispielvezeichnis unter

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

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

4.1 Paketoptionen

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

Die Paketoptionen bestimmen das Verhalten des Gesamtpakets, d. h. aller Befehle.

Die folgenden Optionen sind definiert:

4.1.1	draft	10
4.1.2	final	10
4.1.3	commandnameprefix	10
4.1.4	markup	11
4.1.5	addedmarkup	12
4.1.6	deletedmarkup	12
4.1.7	highlightmarkup	13
4.1.8	commentmarkup	14
4.1.9	authormarkup	14
4.1.10	authormarkupposition	15
4.1.11	authormarkuptext	15
4.1.12	defaultcolor	15
4.1.13	todonotes	16
4.1.14	truncate	16
4.1.15	ulem	16
4.1.16	xcolor	17

4.1.1 draft

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

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`.

4.1.2 final

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

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`.

4.1.3 commandnameprefix

```
\usepackage[commandnameprefix=<strategy>]{changes}
```

Die *commandnameprefix*-Option legt die Präfix-Strategie für die Hervorhebungskommandos fest. Das ist hilfreich, wenn ein anderes Paket bereits ein Kommando definiert hat, z. B. `\comment` oder `\highlight`.

Standardmäßig wird in diesem Fall ein Fehler ausgegeben und kein Präfix vergeben (Option ist nicht oder auf *none* gesetzt).

Wenn eine Präfix-Strategie angegeben ist, wird dem betreffenden Kommando ein „ch“ vorangestellt. Die Strategie legt fest, welche Kommandos einen Präfix bekommen.

Diese Option wirkt sich nur auf die Änderungs- und Hervorhebungskommandos aus:

- `\added` → `\chadded`
- `\deleted` → `\chdeleted`
- `\replaced` → `\chreplaced`
- `\highlight` → `\chhighlight`
- `\comment` → `\chcomment`

Folgende Strategien stehen zur Verfügung:

- | | |
|-------------------|--|
| <code>none</code> | kein Präfix, wenn ein Kommando bereits definiert ist, wird eine Fehlermeldung ausgegeben (Standardstrategie) |
|-------------------|--|

<code>ifneeded</code>	falls ein Kommando bereits definiert ist, bekommt es ein Präfix und eine Warnung wird ausgegeben. Je nachdem, welche Kommandos bereits definiert sind, wird das Dokument eine Mischung von Kommandos mit und ohne Präfix enthalten. Diese Strategie ist vor allem nützlich, wenn nur die Kommandos <code>\comment</code> oder <code>\highlight</code> bereits definiert sind und für die Änderungsmarkierung die normalen Kommandos <code>\added</code> , <code>\deleted</code> und <code>\replaced</code> genutzt werden sollen.
<code>always</code>	alle Kommandos bekommen ein Präfix, eine entsprechende Nachricht wird ins Log geschrieben

Beispiele

```
\usepackage[commandnameprefix=none]{changes} ~ \usepackage{changes}
\usepackage[commandnameprefix=ifneeded]{changes}
\usepackage[commandnameprefix=always]{changes}
```

4.1.4 markup

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

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`, `deletedmarkup`, `commentmarkup` oder `highlightmarkup` geändert werden.

Die folgenden Werte für *markup* sind definiert:

<code>default</code>	default für zugefügten, gelöschten und hervorgehobenen Text sowie Kommentare (default)
<code>underlined</code>	zugefügter Text wird unterstrichen, gewellt unterstrichen für Hervorhebungen, default für gelöschten Text sowie Kommentare
<code>bfit</code>	fetter zugefügter Text, schräger gelöschter Text, default für hervorgehobenen Text sowie Kommentare
<code>nocolor</code>	es werden keine Farben verwendet, zugefügter Text wird unterstrichen, gewellt unterstrichen für Hervorhebungen, default für gelöschten Text sowie Kommentare

Beispiele

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

Wenn von farbigem zu nichtfarbigem Markup oder umgekehrt gewechselt wird und eine Hilfsdatei existiert werden einige Kompilierfehler angezeigt. Über diese ist hinwegzuspringen, beim zweiten Durchlauf sollten die Fehler verschwunden sein.

4.1.5 addedmarkup

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

Die `addedmarkup`-Option wählt ein vordefiniertes visuelles Markup für zugefügten Text. Das default-Markup wird gewählt, wenn die Option nicht gesetzt wird. Die Option `addedmarkup` überschreibt das mit `markup` gewählte Markup.

Die folgenden Werte für `addedmarkup` sind definiert:

<code>colored</code>	kein Textmarkup, nur farbige Kennzeichnung – Beispiel (default)
<code>uline</code>	unterstrichener Text – <i>Beispiel</i>
<code>uuline</code>	doppelt unterstrichener Text – <i>Beispiel</i>
<code>uwave</code>	gewellt unterstrichener Text – <i>Beispiel</i>
<code>dashuline</code>	gestrichelt unterstrichener Text – <i>Beispiel</i>
<code>dotuline</code>	gepunktet unterstrichener Text – <i>Beispiel</i>
<code>bf</code>	fetter Text – Beispiel
<code>it</code>	italic Text – <i>Beispiel</i>
<code>sl</code>	schräger Text – <i>Beispiel</i>
<code>em</code>	hervorgehobener Text – <i>Beispiel</i>

Die Ausgabe ersetzten Texts ist ein Kombination von zugefügtem und gelöschten Text, daher beeinflusst deren Layoutänderung auch das Layout ersetzen Texts.

Beispiele

```
\usepackage[addedmarkup=colored]{changes} ~ \usepackage{changes}
\usepackage[addedmarkup=uline]{changes}
\usepackage[addedmarkup=bf]{changes}
```

4.1.6 deletedmarkup

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

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 für `addedmarkup` sind definiert:

sout	durchgestrichener Text – Beispiel (default)
xout	schräg durchgestrichener Text – Beispiel
colored	kein Textmarkup, nur farbige Kennzeichnung – Beispiel
uline	unterstrichener Text – Beispiel
uuline	doppelt unterstrichener Text – <u>Beispiel</u>
uwave	gewellt unterstrichener Text – <u>Beispiel</u>
dashuline	gestrichelt unterstrichener Text – <u>Beispiel</u>
dotuline	gepunktet unterstrichener Text – <u>Beispiel</u>
bf	fetter Text – Beispiel
it	italic Text – <i>Beispiel</i>
sl	schräger Text – <i>Beispiel</i>
em	hervorgehobener Text – <i>Beispiel</i>

Die Ausgabe ersetzten Texts ist ein Kombination von zugefügtem und gelöschten Text, daher beeinflusst deren Layoutänderung auch das Layout ersetzen Texts.

Beispiele

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

4.1.7 highlightmarkup

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

Die highlightmarkup-Option wählt ein vordefiniertes visuelles Markup für hervorgehobenen Text. Das default-Markup wird gewählt, wenn die Option nicht gesetzt wird. Die Option highlightmarkup überschreibt das mit markup gewählte Markup.

Die folgenden Werte für *highlightmarkup* sind definiert:

background	Hervorhebung durch Hintergrundfarbe – Beispiel (default)
uuline	doppelt unterstrichener Text – <u>Beispiel</u>
uwave	gewellt unterstrichener Text – <u>Beispiel</u>

Beispiele

```
\usepackage[highlightmarkup=background]{changes} ~ \usepackage{
changes}
\usepackage[highlightmarkup=uuline]{changes}
```

4.1.8 commentmarkup

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

Die *commentmarkup*-Option wählt ein vordefiniertes visuelles Markup für Kommentare. Das default-Markup wird gewählt, wenn die Option nicht gesetzt wird. Die Option *commentmarkup* überschreibt das mit *markup* gewählte Markup.

Die folgenden Werte für *commentmarkup* sind definiert:

Beispielkommentar	<code>todo</code>	Kommentar als ToDo-Notiz, die nicht in der Liste der ToDos erscheint (default)
Beispielkommentar	<code>margin</code>	Kommentar im Seitenrand
	<code>footnote</code>	Kommentar als Fußnote ¹
	<code>uwave</code>	gewellt unterstrichener Text – <u>Beispielkommentar</u>

Beispiele

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

4.1.9 authormarkup

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

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 für *authormarkup* sind definiert:

<code>superscript</code>	hochgestellter Text – Text ^{Autor} (default)
<code>subscript</code>	tiefgestellter Text – Text _{Autor}
<code>brackets</code>	Text in Klammern – Text(Autor)
<code>footnote</code>	Text in einer Fußnote – Text ²
<code>none</code>	keine Autor-Identifizierung

Beispiele

```
\usepackage[authormarkup=superscript]{changes} ~ \usepackage{changes}
\usepackage[authormarkup=brackets]{changes}
\usepackage[authormarkup=none]{changes}
```

¹ Beispielkommentar

² Autor

4.1.10 authormarkupposition

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

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 für *authormarkupposition* sind definiert:

<code>right</code>	rechts vom Text – Text ^{Autor} (default)
<code>left</code>	links vom Text – ^{Autor} Text

Beispiele

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

4.1.11 authormarkuptext

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

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 für *authormarkuptext* sind definiert:

<code>id</code>	Autoren-ID – Text ^{ID} (default)
<code>name</code>	Autorenname – Text ^{Autorenname}

Beispiele

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

4.1.12 defaultcolor

```
\usepackage[defaultcolor=<color>]{changes}
```

Die *defaultcolor*-Option legt die Standardfarbe für Autoren, insbesondere den anonymen Autor fest. Sie können die Farben des *xcolor*-Pakets benutzen.

Die Standardfarbe ist blau (*blue*).

Beispiele


```
\usepackage[defaultcolor=blue]{changes} ~ \usepackage{changes}  
\usepackage[defaultcolor=magenta]{changes}
```

4.1.13 todonotes

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

Optionen für das *todonotes*-Paket können als Parameter der *todonotes*-Option angegeben werden. Mehrere Optionen oder Angaben mit Sonderzeichen müssen in geschweifte Klammern gesetzt werden.

Beispiele

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

4.1.14 truncate

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

Optionen für das *truncate*-Paket können als Parameter der *truncate*-Option angegeben werden. Mehrere Optionen oder Angaben mit Sonderzeichen müssen in geschweifte Klammern gesetzt werden.

Beispiele

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

4.1.15 ulem

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

Optionen für das *ulem*-Paket können als Parameter der *ulem*-Option angegeben werden. Mehrere Optionen oder Angaben mit Sonderzeichen müssen in geschweifte Klammern gesetzt werden.

Beispiele

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

4.1.16 xcolor

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

Optionen für das *xcolor*-Paket können als Parameter der *xcolor*-Option angegeben werden. Mehrere Optionen oder Angaben mit Sonderzeichen müssen in geschweifte Klammern gesetzt werden.

Beispiele

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

4.2 Änderungsmanagement

4.2.1	\added	17
4.2.2	\deleted	18
4.2.3	\replaced	18

4.2.1 \added

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

Der Befehl `\added` markiert zugefügten Text. Der neue Text wird in geschweiften Klammern übergeben.

Das optionale Argument enthält Key-Value-Paare für die Angabe von Autor-ID sowie eines Kommentars. Die Autor-ID muss mit einer mit dem `\definechangesauthor`-Befehl definierten ID übereinstimmen. Enthält der Kommentar Sonderzeichen oder Leerzeichen, ist er in geschweifte Klammern einzuschließen.

Wenn ein Kommentar angegeben wurde, wird das direkte Autormarkup am geänderten Text unterdrückt, da es im Kommentar erscheint.

Beispiele

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

Resultat[EK 3] has
to be in itThis is **new** text. This is **new**^{EK} text too. This is **more new** text. This is the last **new** text.

[1] anonymous

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

Der Befehl `\deleted` markiert gelöschten Text. Der gelöschte Text wird in geschweiften Klammern übergeben.

Optionale Argumente: siehe `\added` (Abschnitt 4.2.1).

Beispiele

```
This is \deleted{old} text.
This is \deleted[id=EK]{old} text too.
This is more \deleted[id=EK, comment={too old}]{old} text.
This is the last \deleted[comment=away]{old} text.
```

Resultat[EK 4] too
oldThis is **old** text. This is **old**^{EK} text too. This is **more old** text. This is the last **old** text.

[2] away

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

Der Befehl `\replaced` markiert geänderten Text. Der neue sowie der alte Text werden in dieser Reihenfolge jeweils in geschweiften Klammern übergeben.

Optionale Argumente: siehe `\added` (Abschnitt 4.2.1).

Die Ausgabe ersetzten Texts ist ein Kombination von zugefügtem und gelöschten Text, daher beeinflusst deren Layoutänderung auch das Layout ersetzen Texts.

Beispiele

```
This is \replaced{new}{replaced} text.
This is \replaced[id=EK]{new}{replaced} text too.
This is more \replaced[id=EK, comment={better}]{new}{replaced} text
.
This is the last \replaced[comment=improved]{new}{replaced} text.
```

Resultat

This is **newreplaced** text. This is **newreplaced^{EK}** text too. This is more **newreplaced** text. This is the last **newreplaced** text.

[EK 5] better

[3] improved

4.3 Hervorhebungen und Kommentare

4.3.1 `\highlight` 19

4.3.2 `\comment` 19

4.3.1 `\highlight`

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

Der Befehl `\highlight` markiert hervorgehobenen Text. Der hervorzuhebende Text wird in geschweiften Klammern übergeben.

Optionale Argumente: siehe `\added` (Abschnitt 4.2.1).

Beispiele

```
This is \highlight{highlighted} text.
This is \highlight[id=EK]{highlighted} text too.
This is more \highlight[id=EK, comment={Good one.}]{highlighted}
text.
This is the last \highlight[comment=remember]{highlighted} text.
```

Resultat

This is **highlighted** text. This is **highlighted^{EK}** text too. This is more **highlighted** text. This is the last **highlighted** text.

[EK 6] Good one.

[4] remember

4.3.2 `\comment`

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

Der Befehl `\comment` fügt dem Dokument einen Kommentar hinzu. Der Kommentar wird als in geschweiften Klammern übergeben.

Der Befehl besitzt nur ein optionales Argument: ein Key-Value-Paar für die Angabe der Autor-ID. Die Autor-ID muss mit einer mit dem `\definechangesauthor`-Befehl definierten ID übereinstimmen.

Die Kommentare werden durchnummeriert, die Nummer erscheint im Kommentar.

Beispiele

```
This is \comment{Sure}commented text.  
This is \comment[id=EK]{Correct.}commented text too.
```

Resultat

[5] Sure

This is commented text. This is commented text too.

[EK 7] Cor-
rect.

4.4 Änderungsübersicht

4.4.1 \listofchanges

```
\listofchanges[style=<style>, title=<title>, show=<type>]
```

Der Befehl `\listofchanges` gibt eine Liste oder Zusammenfassung 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.

Es können drei optionale Argumente angegeben werden:

<code>style</code>	Listenstil
<code>title</code>	individueller Titel
<code>show</code>	Änderungstypen

style Über das Argument `style` können verschiedene Listenstile für die Anzeige ausgewählt werden. Es sind folgende drei Stile definiert:

<code>list</code>	gibt die Änderungsliste wie ein Inhaltsverzeichnis aus (default)
<code>summary</code>	gibt die Anzahl der Änderungen gruppiert nach Autor aus
<code>compactsummary</code>	wie <code>summary</code> , jedoch werden Änderungen mit Anzahl 0 nicht ausgegeben

title Mit dem Argument `title` kann ein eigener Titel für die Änderungsliste angegeben werden. Wenn Sie Sonderzeichen oder Leerzeichen im Titel benutzen wollen, klammern Sie den Titel geschweift ein.

show Das Argument `show` gibt an, welche Änderungstypen in der Änderungsliste ausgegeben werden. Sie können die Typen mit Hilfe des Zeichens `|` kombinieren. Wenn Sie z. B. alle neuen Texte und alle Löschungen anzeigen wollen, geben Sie `show=added|deleted` an.

Die folgenden Werte sind definiert:

<code>all</code>	alle Typen (default)
<code>added</code>	nur neue Texte
<code>deleted</code>	nur Löschungen
<code>replaced</code>	nur Ersetzungen
<code>highlight</code>	nur Hervorhebungen
<code>comment</code>	nur Kommentare

Beispiele

```
\listofchanges
\listofchanges[style=list] ~ \listofchanges
\listofchanges[style=summary, title={My Summary}]
\listofchanges[title={List of comments}, show=comment]}
\listofchanges[style=compactsummary, show=added|deleted|replaced,
  title={Text changes}]}
```

4.5 Autorenverwaltung

4.5.1 `\definechangesauthor`

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

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 blau genutzt.

Der Name wird in der Änderungsliste sowie im Markup benutzt, im Markup jedoch nur, wenn die entsprechende Option gesetzt ist.

Das Paket definiert einen anonymen Autor vor, der ohne ID genutzt werden kann.

Beispiele

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

4.6 Anpassung der Ausgabe

4.6.1	Werte für die Anpassung in den Makros	22
4.6.2	<code>\setaddedmarkup</code>	23
4.6.3	<code>\setdeletedmarkup</code>	23
4.6.4	<code>\sethighlightmarkup</code>	24
4.6.5	<code>\setcommentmarkup</code>	24
4.6.6	<code>\setauthormarkup</code>	25
4.6.7	<code>\setauthormarkupposition</code>	25
4.6.8	<code>\setauthormarkuptext</code>	25
4.6.9	<code>\settruncatewidth</code>	26
4.6.10	<code>\setsummarywidth</code>	26
4.6.11	<code>\setsummarytowidth</code>	26
4.6.12	<code>\setlocextension</code>	27
4.6.13	<code>\setsoextension</code>	27

4.6.1 Werte für die Anpassung in den Makros

Wenn Sie die Ausgabe anpassen wollen, können sie beliebige \LaTeX -Befehle sowie spezielle Werte oder Makros des *changes*-Pakets benutzen. Einige Werte sind makrospezifisch, diese sind in den Makros beschrieben.

Die folgenden Werte und Makros können Sie in jedem Kommando benutzen:

- beliebige \LaTeX -Befehle
- Autorenfarbe wird in der Farbe „authorcolor“ bereitgestellt
- boolescher Test auf farbigen Text mit „`\IfIsColored`“

Ich stelle nicht den vollen Zugriff auf alle Parameter des Markups zur Verfügung, um die Makros einfach zu halten. So ist z. B. die Autoren-ID nur für `\setcommentmarkup` zugreifbar.

Die Ausgabe ersetzten Texts ist ein Kombination von zugefügtem und gelöschten Text.

4.6.2 `\setaddedmarkup`

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

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 die Definition:

- neuer Text wird mit „#1“ gesetzt

Beispiele

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

Die Ausgabe ersetzten Texts ist ein Kombination von zugefügtem und gelöschten Text, daher beeinflusst deren Layoutänderung auch das Layout ersetzen Texts.

4.6.3 `\setdeletedmarkup`

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

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

Werte für die Definition:

- gelöschter Text wird mit „#1“ gesetzt

Die Ausgabe ersetzten Texts ist ein Kombination von zugefügtem und gelöschten Text, daher beeinflusst deren Layoutänderung auch das Layout ersetzen Texts.

Beispiele

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


4.6.4 \sethighlightmarkup

`\sethighlightmarkup{<definition>}`

Der Befehl `\sethighlightmarkup` legt fest, wie hervorgehobene Texte gesetzt werden. Ohne andere Definition gilt, dass die Hervorhebung über die Hintergrundfarbe erfolgt oder je nach Option `markup` bzw. `commentmarkup` erscheint.

Werte für die Definition:

- hervorgehobener Text wird mit „#1“ gesetzt

Beispiele

```
\sethighlightmarkup{\emph{#1}}
\sethighlightmarkup{{\IfIsColored{\color{authorcolor}}{}}|: #1}}
```

4.6.5 \setcommentmarkup

`\setcommentmarkup{<definition>}`

Der Befehl `\setcommentmarkup` legt fest, wie Kommentare gesetzt werden. Ohne andere Definition gilt, dass Kommentare im Rand oder je nach Option `markup` bzw. `commentmarkup` erscheint.

Werte für die Definition:

- Kommentar wird mit „#1“ gesetzt
- Autor-ID steht in „#2“ bereit
- Autor-Ausgabe (ID oder Name) wird mit „#3“ gesetzt
- Kommentaranzahl steht in „authorcommentcount“ bereit
- boolscher Test auf anonymen Autor mit „\IfIsAnonymous“

Beispiele

```
\setcommentmarkup{-- #1 --}
\setcommentmarkup{{\IfIsColored{\color{authorcolor}}{}}#1}}
\setcommentmarkup{\IfIsAnonymous{#2}}{\textbf{#3: }}#1}
\setcommentmarkup{[\arabic{authorcommentcount}] #1}
```

4.6.6 \setauthormarkup

`\setauthormarkup{<definition>}`

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

Werte für die Definition:

- Autor-Ausgabe (ID oder Name) wird mit „#1“ gesetzt

Beispiele

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

4.6.7 \setauthormarkupposition

`\setauthormarkupposition{<authormarkupposition>}`

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.

Die folgenden Werte für *authormarkupposition* sind definiert:

right	rechts vom Text – Text ^{Autor} (default)
left	links vom Text – ^{Autor} Text

Beispiele

```
\setauthormarkupposition{right}  
\setauthormarkupposition{left}
```

4.6.8 \setauthormarkuptext

`\setauthormarkuptext{<authormarkuptext>}`

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

Die folgenden Werte für *authormarkuptext* sind definiert:

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

Beispiele

```
\setauthormarkuptext{id}  
\setauthormarkuptext{name}
```

4.6.9 \settruncatewidth

```
\settruncatewidth{<width>}
```

Der Befehl `\settruncatewidth` legt die Breite der Textkürzung in der Änderungsliste fest. Die Standardbreite ist `0.6\textwidth`.

Beispiele

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

4.6.10 \setsummarywidth

```
\setsummarywidth{<width>}
```

Der Befehl `\setsummarywidth` legt die Breite der Änderungsliste mit Stil `summary` bzw. `compactsummary` fest. Die Standardbreite ist `0.3\textwidth`.

Beispiele

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

4.6.11 \setsummarytewidth

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

Der Befehl `\setsummarytewidth` legt die Breite der Änderungsliste mit Stil `summary` bzw. `compactsummary` anhand der Breite des übergebenen Texts fest.

Beispiele

```
\setsummarytewidth{Highlighted \qqquad}  
\setsummarytewidth{The longest text imaginable for the summary.}
```

4.6.12 `\setlocextension`

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

Der Befehl `\setlocextension` legt die Dateierweiterung der Hilfsdatei für die Liste der Änderungen (loc-Datei³) fest. Ohne andere Definition gilt das Suffix „loc“.

Im angegebenen Beispiel würde für „foo.tex“ eine Hilfsdatei erzeugt werden, die „foo.listofchanges“ bzw. „foo.lochg“ statt des Standardnamens „foo.loc“ hieße.

Beispiele

```
\setlocextension{listofchanges}  
\setlocextension{lochg}
```

Nutzen Sie keine Standard- \TeX -Dateierweiterungen wie „toc“ oder „lof“, da das den normalen \TeX -Lauf stören würde.

4.6.13 `\setsocextension`

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

Der Befehl `\setsocextension` legt die Dateierweiterung der Hilfsdatei für die Änderungszusammenfassung (soc-Datei⁴) fest. Ohne andere Definition gilt das Suffix „soc“.

Im angegebenen Beispiel würde für „foo.tex“ eine Hilfsdatei erzeugt werden, die „foo.changes“ bzw. „foo.chg“ statt des Standardnamens „foo.soc“ hieße.

Beispiele

```
\setsocextension{changes}  
\setsocextension{chg}
```

Nutzen Sie keine Standard- \TeX -Dateierweiterungen wie „toc“ oder „lof“, da das den normalen \TeX -Lauf stören würde.

³ „loc“ steht dabei für „list of changes“.

⁴ „soc“ steht dabei für „summary of changes“.

4.7 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:

etoolbox	stellt verbesserte <code>\if</code> -Abfragen, <i>bools</i> oder Listenoperationen zur Verfügung
truncate	Kürzung von Texten (für die Änderungsliste)
xkeyval	Eingabe von Optionen mit Werteübergabe
xstring	verbesserte Stringoperationen

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

todonotes	wird geladen, wenn Kommentare als ToDo-Notizen gesetzt werden (default Markup)
ulem	wird geladen, wenn Text durchgestrichen, gewellt markiert oder ausge-x-t wird (default Markup)
xcolor	wird geladen, wenn farbiger Text genutzt wird (default Markup)

5 Markierungen aus den Dateien entfernen

Die Entfernung der Markierungen aus dem Quelltext müssen Sie von Hand vornehmen, dafür steht auch ein Script von Yvon Cui zur Verfügung. Das Script liegt im Verzeichnis:

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

Das Script entfernt alle Markierungen, indem die Änderungen angenommen oder abgelehnt werden. Sie können die zu entfernenden Markierungen individuell im interaktiven Modus selektieren bzw. selektieren, indem Sie das Skript ohne Optionen starten.

Das Skript benötigt *python3*.

Nutzen Sie das Skript wie folgt:

```
python pyMergeChanges.py [-arh] <Input File> <Output File>
```

Options:

```
-a: accept all added, deleted and replaced  
-r: reject all added, deleted and replaced  
-h: remove all highlights
```

If no option is given, runs interactively.

Starten Sie das Skript ohne Optionen und Dateien für eine kurze Hilfe:

```
python pyMergeChanges.py
```

Bekannte Probleme:

- entfernt nur Markierungen, die in einer Zeile stehen, Markierungen, die mehrere Zeilen umfassen, werden ignoriert

6 Bekannte Probleme und Lösungen

In diesem Kapitel sammle ich die häufigsten Probleme und mir dazu bekannte Lösungen. Wenn Ihr Problem hier nicht aufgeführt ist, sehen Sie bitte im Issue-Tracker auf gitlab nach, ob das Problem dort beschrieben ist (es gibt eine Suche):

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

Wenn das alles zu nichts führt, öffnen Sie bitte ein neues Issue für das Problem, beschreiben Sie das Problem genau und liefern Sie, wenn möglich, eine kleine Beispieldatei mit dem problematischen Verhalten mit.

6.1 Besondere Inhalte

Die Änderungsmarkierung von Text funktioniert recht gut, es können auch ganze Absätze markiert werden. Die Markierung ist eingeschränkt oder nicht möglich für:

- Abbildungen
- Tabellen
- Überschriften
- manche Kommandos
- mehrere Absätze (manchmal)

Sie können versuchen, solchen Text in eine eigene Datei auszulagern, und diese mit `input` einzubinden. Manchmal hilft das, oft ist es einen Versuch wert. Danke an Charly Arenz für diesen Tip.

6.2 Fußnoten und Randnotizen

Fußnoten oder Randnotizen werden in bestimmten Umgebungen, z. B. Tabellen oder der *tabbing*-Umgebung, nicht korrekt gesetzt. Vermeiden Sie das Markup, wenn Sie diese Umgebungen benutzen.

6.3 Das *ulem*-Paket

Ich verwende standardmäßig das *ulem*-Paket für das Durchstreichen von Text. Das führt bei manchen Befehlen und Umgebungen zu Problemen, z. B.

- im Mathemodus
- bei Verwendung des *siunitx*-Pakets
- bei Nutzung der `\citet`- oder `\citep`-Befehle

In dem Fall gibt es wenig gute Möglichkeiten, am besten ist es, das Markup für Löschungen selbst zu definieren und das *ulem*-Paket zu vermeiden. Siehe

- Abschnitt 4.1.6
- Abschnitt 4.6.3

6.4 Kommando bereits definiert – command already defined

Einige Pakete benutzen für ihre Kommandos dieselben Namen wie das *changes*-Paket, insbesondere `\comment` und `\highlight` sind keine originell benannten Kommandos.

In diesem Fall kann *changes* seinen Kommandos ein Präfix voranstellen, um Namenskollosionen zu vermeiden. Das wird über die Option `commandnameprefix` gesteuert, die in Abschnitt 4.1.3 beschrieben ist.

Damit das funktioniert, muss das *changes*-Paket nach dem Paket geladen werden, das das Kommando auch benutzt.

7 Autorinnen und Autoren

Am *changes*-Paket haben mehrere Autorinnen und Autoren mitgewirkt. Viele Probleme wurden in `de.comp.text.tex` gelöst oder deren Lösung durch Lösungsansätzen inspiriert. Danke.

Die Autorinnen und Autoren sind in alphabetischer Reihenfolge:

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

8 Versionen

Für eine Liste der verfügbaren Versionen und deren Änderungen gehen Sie bitte zu <https://gitlab.com/ekleinod/changes/blob/master/changelog.md>

Dort sind auch die bereits implementierten aber noch nicht veröffentlichten Änderungen verzeichnet.

Wenn Sie an geplanten, zukünftigen Änderungen interessiert sind, finden Sie diese unter <https://gitlab.com/ekleinod/changes/milestones>

9 Weitergabe, Copyright, Lizenz

Copyright 2007-2021 Ekkart Kleinod (ekleinod@edgesoft.de)

Dieses Paket darf unter der „ \LaTeX 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 \LaTeX -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/pyMergeChanges.py
```

und den generierten Dateien

```
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
```