

Makropaket changes

Knut Lickert

14. Juni 2005

Zusammenfassung

Dieses Makropaket stellt das Kommando `change` zur Verfügung. Mit diesem Kommando können Änderungen in einem Dokument gekennzeichnet werden. Das Makro entspricht dem gleichnamigen Makro aus der ltxdoc-Klasse, ist aber in jedem Programm verwendbar.

Offen:

- Datum flexibel (iso+tex)
- Kommando zum ausschalten margins.

1 Kurzdoku

1.1 Makros

`\change` Im Text verwenden zur Definition einer Änderung. Parameter:

1. Version des Dokuments oder Modifikationsnummer o.ä.
2. Ziel der Änderung (optional)
3. Datum im Format JJJJ-MM-TT oder JJJJ/MM/TT
4. Erläuterung der Änderung.

Version, Datum und Erläuterung sollten selbsterklärend sein.

Das Ziel (target) der Änderung ermöglicht es, eine Art „Aktivierungskontrolle“ zu ermöglichen.

`\reportChangesFrom` Angabe des Datums (JJJJ-MM-TT), ab dem die Änderungen gezeigt werden sollen. Default ist 0000-12-31 (d.h. alle Änderungen).

Sinn des Parameters: Ganz alte Änderungen sind irgendwann nicht mehr interessant, ich will meistens nur die Änderungen der letzten Monate zeigen, nicht alle Änderungen.

`\reportChangesTo` gibt es nicht. Wozu auch?

`\listofchanges` Ausgabe aller Änderungen (in Abhängigkeit von `\reportChangesFrom`).

`changeslist` Umgebung, die `\change`-Makros aufnehmen kann. Ergebnis ist eine Liste aller Änderungen in einer Tabelle, Je nach Menge der Änderungen, bietet sich eine longtable-Umgebung an. Dazu sollte die Option longtable angegeben werden.

1.2 Optionen

<code>silent</code>	Es werden keine Nachrichten im Log ausgegeben.
<code>index</code>	Die Änderungsnummer und Änderungsdatum werden in einem eigenen Index ausgegeben.
<code>margin</code>	Die Änderungsnummer und Änderungsdatum werden als Randnotiz ausgegeben.
<code>longtable</code>	in Der Umgebung <code>changeslist</code> werden die Änderungen in einer longtable-Umgebung ausgegeben.

1.3 Anpassungen

Das Makro `\change` ruft verschiedene Makros auf, die verschiedene Ausgaben erzeugen. Sollen bestimmte Ausgaben erzeugt werden, können diese Makros mit `\renewcommand` angepasst werden. Die folgenden Makros haben alle vier Parameter:

1. Version, Modifikations- oder Transportnummer
2. Ziel (optionale Parameter von `\change`)
3. Datum
4. Beschreibung

<code>\changesIndex</code>	Sollen Änderungen als Index-Einträge ausgegeben werden, kann das Makro <code>\changesIndex</code> undefiniert werden. Mit Option „index“ (default) wird in einen eigene Index „Changes“ geschrieben. Siehe Dokumentation <code>index.sty</code> oder den Code.
<code>\changesMargin</code>	Sollen Änderungen als Randnotiz ausgegeben werden, kann das Makro <code>\changesMargin</code> undefiniert werden. Per Standard ist das Makro leer, Option „margin“ setzt eine Ausgabe mit Versionsnumemr und Datum.
<code>changeslist</code>	Lokale Umdefinition des <code>\changes</code> -Kommandos zur Ausgabe in einer Liste. Geeignet für Zusammenfassungen. Es erfolgt kein Eintrag in den Index. Mit der Option <code>longtable</code> wird statt einer Liste eine longtable-Umgebung verwendet.

2 Beispiele

2.1 Dokumentversion

Ein Dokument hat verschiedene Versionen. Jede Version erhält eine Nummer, eine Versionsverwaltung ist möglich.

2.2 Softwaredokumentation

Eine Anwendung soll beschrieben werden. Die Anwendung besteht aus verschiedenen Programmen, Datenbankstrukturen. . . und wird auf verschiedenen Systemen verwendet. Wann welche Funktion auf welchem System aktiv wurde soll nachvollziehbar sein.

Statt einer Versionsnummer pro Programm wird für jede Änderung (die verschiedene Programme betreffen kann) eine Modifikationsnummer erzeugt. Im Programmcode wird jede Änderung mit der Modifikationsnummer referenziert.

In der Programmdokumentation wird jede Änderung in Bezug zur Modifikationsnummer gesetzt. Beispiel:

```
\change{12}{2004-09-01}{Prog. X: Neue Funktion Y}
```

12 ist dabei eine beliebige Modifikationsnummer, die alle Änderungen zusammenfasst (kann z.B. ein Teil-Projekt o.ä. sein). Ja nach System wird von der Entwicklungsumgebung eine automatische Transportnummer erzeugt die verwendet werden kann.

Werden alle Änderungen der Modifikation (Transport, jar-File oder was immer man hat) in ein Zielsystem transportiert (Testsystem, Produktivsystem, Rechner von Person X. . .) so wird das auch protokolliert:

```
\change{12}[Rechner X]{2004-09-01}{Mod. 12: Änderungen gemäß Projekt X}
```

3 Coding

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{changes}[\filedate\space\fileversion\space changes]
3 \RequirePackage{isodate}
4 \RequirePackage{index2}
```

Option silent: No message is given in the log-File.

```
5 \newif\ifchg@silent
6 \chg@silentfalse
7 \DeclareOption{silent}{\chg@silenttrue}
```

Fixme: *Bug index.sty*

Bug in index.sty. Es wird nach dem Paket article fix abgefragt.

```
8 \providecommand{\@makeschapterhead}{\relax}
```

`\chg@analysedate` The given date is analysed. It doesn't matter, if the date is in format yyyy-mm-dd or yyyy/mm/dd

```
9 \def\chg@analysedate#1#2#3#4#5{#1#2#3#4\chg@analysedateII}%Jahr
10 \def\chg@analysedateII#1#2#3#4#5/{#1#2#4#5}
```

`\chg@analysedateiso` The given date is analysed. It doesn't matter, if the date is in format yyyy-mm-dd or yyyy/mm/dd. Result is a iso-date (used by printdate)

```
11 \def\chg@analysedateiso#1#2#3#4#5{#1#2#3#4\chg@analysedateisoII}
12 \def\chg@analysedateisoII#1#2#3#4#5{-#1#2-#4#5}
```

`\chg@changefromdate` Save the date, from which the changes should be printed.

```

13 \newcommand{\chg@changefromdate}{0000-12-31}
14 %\newcommand{\chg@changetodate}{99991231}

    Two counter to save dates a numbers (YYYYMMDD).
15 \newcounter{chg@checkdate}
16 \newcounter{chg@changefromdate}

```

`\reportChangesFrom` User command to change the date, from which the changes should be reported.

```

17 \newcommand{\reportChangesFrom}[1]{%
18     \renewcommand{\chg@changefromdate}{#1}%
19     \setcounter{chg@changefromdate}{\chg@analysedate#1/}%
20 }

```

`\changes` User command to define a change. Parameters: Version, [optional target], date, description. If you use this command with doc.sty, you don't need to change the parameters. Target is added.

```

21 \newcommand{\changes}[1]{%{version/Transport}[target]{date}{Text}
22     \gdef\@changesversion{#1}
23     \@changesII
24 }

```

`\@changesII` Second part of `\changes`. Optional parameter for the target and additional text.

```

25 \newcommand{\@changesII}[3] [] {%[target]{date}{Text}
26     \bgroup%
27     \setcounter{chg@checkdate}{\chg@analysedate#2/}%
28     \ifnum \arabic{chg@checkdate} > \arabic{chg@changefromdate}%
29         \changesIndex{\@changesversion}{#1}{#2}{#3}%
30         \changesMargin{\@changesversion}{#1}{#2}{#3}%
31     \else%
32         \ifchg@silent\else\PackageInfo{changes.sty}{supressed change \@changesversion #1 #2 #3}
33     \fi%
34     \egroup%
35 }

```

`\changesIndex` Internal macro to write the change to the index. It must be activated by the option "index". Can be redefined by a user. The four parameter are:

1. Version, Transport number
2. Target (optional parameter form `\change`
3. date
4. description

```

36 \newcommand{\changesIndex}[4] {}%{version/Transport}{target}{date}{Text}
37 \DeclareOption{index}{
38 \newindex{Changes}{Changesidx}{Changesind}{Changes}
39 \renewcommand{\changesIndex}[4] {}%{version/Transport}{date}{Text}

```

```

40 %      \index[Changes]{#3\printdate{\chganalysedateiso#3}!#4}%
41 \ifthenelse{\equal{#2}}{%
42   \index[Changes]{#3\printdate{\chganalysedateiso#3}!#4}%
43   }{%
44   \index[Changes]{#3\printdate{\chganalysedateiso#3}!#4 (\(\to\)#2)}%
45   }
46 } }
47 \ExecuteOptions{index}

```

`\listofchanges` Write the list of all changes. This requires a run with `makeindex`.

```

48 \newcommand{\listofchanges}{
49 %   \renewcommand{\indexname}{Änderungen (ab \printdate{\chg@changefromdate})}
50   \IfFileExists{\jobname.Changesind}{
51     \printindex[Changes]
52   }{
53     \PackageWarning{Changes}{No Changesind-File}
54   }
55 }

```

`\changesMargin` Internal macro to write the change to the margin. Can be redefined by a user. The four parameter are:

1. Version, Transport number
2. Target (optional parameter form `\change`)
3. date
4. description

```

56 \newcommand{\changesMargin}[4]{%{version/Transport}{target}{date}{Text}
57 \DeclareOption{margin}{
58 \renewcommand{\changesMargin}[4]{%{version/Transport}{date}{Text}
59   \marginpar{\footnotesize #1 \isodate{#3}}%
60 }}

```

Fixme: *changeslist: translation*

`\textchangeslist`

```

61 \newcommand{\textchangeslist}{Transporte}
62 \addto\extrasenglish{\def\textchangeslist{Transports}}
63 \addto\extrasgerman{\def\textchangeslist{Transporte}}
64 \addto\extrasngerman{\def\textchangeslist{Transporte}}

```

Fixme: *changeslist: add to index (optional)*

```

65 \newenvironment{changeslist}{%
66   \textchangeslist%
67   \begin{description}
68   \renewcommand{\@changesII}[3] []{%[target]{date}{Text}
69     \item[ \@changesversion]
70       \mbox{\printdate{\chganalysedateiso##2}$\to$~##1}%
71       \ifthenelse{\equal{##3}}{\par ##3}
72   }

```

```

73   }{
74   \end{description}
75   }

76 \DeclareOption{longtable}{
77 \AtEndOfPackage{\RequirePackage{longtable}}
78 \renewenvironment{changeslist}{%
79   \renewcommand{\@changesII}[3] []{%\[target]{date}{Text}
80     \@changesversion &
81 %     \printdate{\chganalysedateiso##2}&
82     ##2&
83     ##1&
84     ##3\\
85   }
86   \begin{longtable}{\llp{0.5\textwidth}}
87   \multicolumn{4}{c}{\textchangeslist}\hline
88   \endhead
89   }{
90   \end{longtable}
91   }
92 }

```

4 Endanweisungen

```
93 \ProcessOptions\relax
```

Verzeichnis der Korrekturen

FixMe: Bug index.sty	3
FixMe: changeslist: translation	5
FixMe: changeslist: add to index (optional)	5