
Elenco del peccati degli utenti di L^AT_EX 2_ε

ovvero Comandi e pacchetti obsoleti e altri errori

Marc Ensenbach, Mark Trettin
e-mail: Marc.Ensenbach@post.rwth-aachen.de

Versione 2.3 del 20 settembre 2011

Traduzione italiana dall'originale
tedesco di Mauro Sacchetto
e-mail: mauro.sacchetto@gmail.com

Sommario

Questa è la traduzione italiana di «l2tabu».

Essa corrisponde in larga misura al più recente originale tedesco e se ne discosta significativamente, a parte modifiche e aggiunte minori, solo per la sezione 3.4 a pagina 21, che è stata completamente riscritta. Le parti modificate sono indicate in nota, dietro esplicita richiesta dell'autore. Il traduttore, che si è valso dei rilievi mossi a questo proposito dai membri del G_{IT}, desidera qui ringraziare in particolar modo Claudio Beccari e Tommaso Gordini. Un riconoscimento finale va all'autore, Marc Ensenbach, che ha letto con accuratezza questa versione italiana fornendo ulteriori rilievi. [N. d. T.]

Lo stimolo per scrivere questo breve sommario è sorto da una discussione nel T_EX-Newsgroup di lingua tedesca^a sul perdurante ricorso a pacchetti e comandi obsoleti e «superati».

In questo articolo cerchiamo di indicare gli errori più comuni e soluzioni alternative. Questo compendio non intende rimpiazzare né guide introduttive come l2kurz [10] né le De-TeX-FAQ^b, ma semplicemente offrire un breve sommario.

Saremo grati per ogni suggerimento, miglioria e commento.

^a de.comp.text.tex

^b <http://projekte.dante.de/DanteFAQ/WebHome>

Copyright © 2003–2007 by Mark Trettin, 2009–2011 by Marc Ensenbach.

This material may be distributed only subject to the terms and conditions set forth in the *Open Publication License*, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Le denominazioni software e hardware presenti in questo articolo sono, nella maggior parte dei casi, anche marchi registrati e perciò soggette come tali alle normative vigenti.

Ringraziamo Ralf Angeli, Christoph Bier, Christian Faulhammer, Jürgen Fenn¹, Ulrike Fischer, Ralf Heckmann, Yvon Henel², Yvonne Hoffmüller, David Kastrup, Markus Kohm, Thomas Lotze, Gonzalo Medina Arellano³, Frank Mittelbach, Heiko Oberdiek, Mauro Sacchetto⁴, Walter Schmidt, Uwe Siart, Stefan Stoll, Knut Wenzig, Emanuele Zannarini⁵, e Reinhard Zierke per i trucchi, le osservazioni e le correzioni. Chi fosse stato dimenticato ci spedisca una mail.

¹ Traduzione inglese: <CTAN://info/l2tabu/english/l2tabuen.pdf>

² Traduzione francese: <CTAN://info/l2tabu/french/l2tabufr.pdf>

³ Traduzione spagnola: <CTAN://info/l2tabu/spanish/l2tabues.pdf>

⁴ Traduzione italiana: <CTAN://info/l2tabu/italian/l2tabuit.pdf>

⁵ Traduttore della prima versione italiana, 2005

Indice

1	«Peccati mortali»	5
1.1	a4, e a4wide	5
1.2	Modifiche del layout	5
1.3	Modifiche di pacchetti e classi	5
1.4	Modifica dell'interlinea mediante <code>\baselinestretch</code>	6
1.5	Rientro di capoverso e spaziatura fra paragrafi (<code>\parindent</code> , <code>\parskip</code>)	6
1.6	Separazione delle formule dal testo con <code>\$\$...\$\$</code>	7
1.7	<code>\def</code> vs. <code>\newcommand</code>	8
1.8	Utilizzo di <code>\sloppy</code>	8
2	Comandi, classi e pacchetti obsoleti	10
2.1	Comandi	10
2.1.1	Modifica dello stile del carattere	10
2.1.2	Frazioni matematiche (<code>\over</code> vs. <code>\frac</code>)	10
2.1.3	Centratura con <code>\centerline</code>	11
2.2	Classi e pacchetti	12
2.2.1	<code>scrlettr.cls</code> vs. <code>scrlettr2.cls</code>	12
2.2.2	<code>epsf</code> , <code>psfig</code> , <code>epsfig</code> vs. <code>graphics</code> , <code>graphicx</code>	12
2.2.3	<code>doubleSPACE</code> vs. <code>setspace</code>	12
2.2.4	<code>fancyheadings</code> , <code>scrpage</code> vs. <code>fancyhdr</code> , <code>scrpage2</code>	13
2.2.5	La famiglia di pacchetti <code>caption</code>	13
2.2.6	<code>isolatin</code> , <code>umlaut</code> vs. <code>inputenc</code>	13
2.2.7	<code>t1enc</code> vs. <code>fontenc</code>	15
2.2.8	<code>natdin.bst</code> vs. <code>dinat.bst</code>	15
2.2.9	<code>glossary</code> vs. <code>glossaries</code>	15
2.2.10	<code>S1style</code> e <code>S1units</code> vs. <code>siunitx</code>	15
2.2.11	<code>subfigure</code> e <code>subfig</code> vs. <code>subcaption</code>	16
2.3	Caratteri	16
2.3.1	<code>times</code>	16
2.3.2	<code>mathptm</code>	17
2.3.3	<code>pslatex</code>	17
2.3.4	<code>palatino</code>	17
2.3.5	<code>mathpple</code>	18
2.3.6	<code>utopia</code>	18
2.3.7	Lettere greche matematiche non inclinate	18
2.3.8	<code>euler</code> vs. <code>eulervm</code>	19
2.3.9	<code>ae</code> , <code>aecompl</code> e <code>zefonts</code>	19
3	Miscellanea	20
3.1	Oggetti flottanti – «figure», «table»	20
3.2	L'appendice	20

Indice

3.3	Formule matematiche	20
3.4	L'uso di <code>\graphicspath</code>	21
3.5	Le macro <code>\...name</code>	23
A	Esempio di <code>\sloppy</code>	25
	Riferimenti bibliografici	26

1 «Peccati mortali»

In questa sezione abbiamo raccolto i peggiori errori che si incontrano con allarmante frequenza in `de.comp.text.tex` e che fanno arrossire di rabbia gli esperti di quel newsgroup o che fanno venir loro le lacrime agli occhi ; -)

1.1 a4, e a4wide

Questi pacchetti, *tutti e due*, non dovrebbero essere più usati e vanno definitivamente eliminati dai sorgenti di \LaTeX . Devono essere sostituiti dall'opzione di classe `a4paper` per le classi standard o dall'analoga opzione `paper=a4` per KOMA-Script a partire dalla versione 3. Va considerato che il layout prodotto da questi due pacchetti è più che discutibile da un punto di vista tipografico e che ne esistono varie versioni, incompatibili l'una con l'altra. Perciò non si può mai essere sicuri che, compilato in un altro sistema, il documento risulti uguale.

1.2 Modifiche del layout

I margini di pagina impostati dalle classi standard (`article.cls`, `report.cls`, `book.cls`) appaiono di solito troppo ampi. Se non soddisfano, si può ricorrere alle corrispondenti classi (`scrartcl.cls`, `scrreprt.cls`, `scrbook.cls`) del pacchetto KOMA-Script oppure al pacchetto `typearea`, contenuto nella collezione di KOMA-Script. La relativa documentazione `scrguide` [3] contiene molte altre informazioni.

Se davvero si avesse bisogno di margini di pagina diversi da quelli impostati ad esempio da `typearea`, si dovrebbe ricorrere al pacchetto `geometry` e non produrre modifiche «fatte a mano» con `\oddsidemargin` & C. Il pacchetto `ansize` è obsoleto e il pacchetto `vmargin`, qui un tempo raccomandato, dovrebbe allo stesso modo essere evitato, poiché può produrre effetti collaterali indesiderati⁶.

Infine, non vanno mai modificate le impostazioni regolate da `\hoffset` e da `\voffset`, a meno che non si conosca molto bene il funzionamento interno di \TeX .

1.3 Modifiche di pacchetti e classi

Non si devono mai modificare direttamente le classi di documento (`article.cls`, `scrbook.cls` ecc.) oppure i file di stile (`varioref`, `color`)! O si è in grado di costruirsi una classe propria basata su classi preesistenti o file di stile, oppure è meglio *copiare* classi e stili, modificare le copie e salvarle con un *altro* nome.

Un esempio su come costruire classi basate su classi preesistenti si trova nelle FAQ sulla personalizzazione delle classi di documento⁷.

⁶ Cfr. anche Message-ID: <uirrzd16.fsf@uwe-siart.de>

⁷ <http://projekte.dante.de/DanteFAQ/Dokumentenklasse>

Suggerimento In generale, tali classi e pacchetti aggiuntivi dovrebbero essere installati o nell'albero locale o nell'albero personale (presente nella `$HOME` degli utenti dei sistemi Unix) di \TeX , perché con un aggiornamento della distribuzione \TeX queste modifiche andrebbero perse. Se tali modifiche servono solo per un progetto particolare, ma potrebbero poi essere diffuse ad altri, allora la classe personalizzata può essere inclusa nell'attuale directory del progetto.

1.4 Modifica dell'interlinea mediante `\baselinestretch`

In generale, le modifiche dei parametri dovrebbero essere effettuate al livello più elevato disponibile all'utente. Pertanto una modifica dell'interlinea può essere compiuta da tre livelli:

1. ridefinendo `\baselinestretch`.
2. usando il comando \LaTeX `\linespread{<Fattore>}`;
3. usando il pacchetto `setspace`;

Una ridefinizione di parametri come `\baselinestretch` opera al livello più basso e dovrebbe pertanto essere riservata ai pacchetti.

Un metodo migliore, proprio perché espressamente destinato a questo scopo, è l'utilizzo di `\linespread{<Fattore>}`.

Tuttavia il miglior modo di procedere utilizzare del pacchetto `setspace` che, quando modifica l'interlinea, si preoccupa di mantenere gli spazi corretti nelle note a piè di pagina o negli elenchi, ambienti la cui modifica nella maggior parte dei casi può essere indesiderata.

Se pertanto si ha bisogno di una maggiore interlinea (ad esempio una volta e mezza o il doppio), si ricorra innanzitutto al pacchetto `setspace`. Se invece si volessero semplicemente apportare dei piccoli aggiustamenti per caratteri diversi dal predefinito Computer Modern (come Palatino), allora sarebbe più opportuno usare `\linespread{<Fattore>}`.

Nel caso del Palatino si può utilizzare un aggiustamento pari a `\linespread{1.05}`.

1.5 Rientro di capoverso e spaziatura fra paragrafi (`\parindent`, `\parskip`)

A volte può non essere insensato modificare il rientro della prima riga dei capoversi (regolato dal parametro `\parindent`). Si dovrebbe tuttavia tener presente che:

- il rientro va modificato con una misura che dipende dal carattere (em) e non con una misura assoluta (mm). «Dipendenza dal carattere» qui significa che il rientro si modifica in base non soltanto alla dimensione, ma anche al tipo di carattere, poiché l'ampiezza di `1em` è adattata da ciascun disegnatore di caratteri caso per caso a ciascun tipo di carattere;

- va usata la sintassi di L^AT_EX, perché comporta meno problemi. Ad esempio è più facile da verificare per programmi o script esterni⁸ e più semplice da mantenere per l'utente. E così si può prevenire ogni incompatibilità con altri pacchetti (come calc).

▷ **NON:** `\parindent=1em` **MA:** `\setlength{\parindent}{1em}`

Se non si desiderasse *alcun* rientro, ma una distanza aggiuntiva fra paragrafi come segno di cesura, *non* si dovrebbe utilizzare semplicemente

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{\baselineskip}
```

poiché `\parskip` agisce anche su elenchi, indici e testatine.

Il pacchetto `parskip` e le classi KOMA-Script aggiungono dello spazio per evitare questi effetti collaterali. Sull'utilizzo delle diverse opzioni (`parskip=full`, `parskip=half` ecc.) di KOMA-Script cfr. la [scrguide \[3\]](#). Se si utilizza una delle classi di KOMA-Script *non* si deve caricare in aggiunta il pacchetto `parskip`.

1.6 Separazione delle formule dal testo con `$$...$$`

Per favore no! `$$...$$` è un comando di plain T_EX e dovrebbe essere evitato in L^AT_EX, perché rende inconsistente lo spazio verticale tra le formule matematiche in `display` e il testo circostante (cfr. anche la sezione [3.3 a pagina 20](#), in particolare l'Avvertimento relativo a `displaymath` usato insieme a `amsmath`). Inoltre, l'opzione di classe `fleqn` non funziona più.

▷ **NON:** `$$...$$` **MA:** `\[...\]`
oppure (solo con `amsmath`)
`\begin{equation*}`
...
`\end{equation*}`
oppure (solo senza `amsmath`)
`\begin{displaymath}`
...
`\end{displaymath}`

⁸ Analizzare sintatticamente, scorporare.

1.7 `\def` vs. `\newcommand`

Le macro dovrebbero essere definite *sempre* mediante `\newcommand{<nome>}{definizione}` e *non* con `\def{<nome>}{definizione}`⁹.

Il problema fondamentale di `\def` è che non permette di controllare l'esistenza di una macro definita in precedenza. Questa potrebbe pertanto essere sovrascritta senza alcuna notifica o senza un avvertimento.

Inoltre, è sempre possibile ridefinire le macro già esistenti col comando: `\renewcommand{<name>}{...}`.

Chi sa a sufficienza del *perché* ha bisogno di `\def`, conosce anche i vantaggi e gli svantaggi di tale soluzione e può ignorare tranquillamente questa sezione.

1.8 Utilizzo di `\sloppy`

Il commutatore `\sloppy` non dovrebbe essere usato, e segnatamente in modo globale all'interno del preambolo. Se in alcuni paragrafi sorgessero dei problemi con la sillabazione, allora:

1. verificare se sono stati impostati i corretti strumenti di sillabazione (ad esempio mediante `babel`) e i caratteri T1 (cfr. anche le FAQ sulla sillabazione¹⁰);
2. usare il pacchetto `microtype` che adatta leggermente l'ampiezza delle lettere per produrre una migliore divisione in sillabe. Questo adattamento automatico è possibile solo con `pdfTeX` a partire dalla versione 1.20 e funziona solo per la creazione di un file PDF, non invece con l'utilizzo di `pdfTeX` nella modalità DVI;
3. riformulare il testo. Non è assolutamente necessario modificare la frase in cui è presente il problema di sillabazione: spesso è sufficiente riformulare o spostare le frasi precedenti o quelle successive;
4. modificare leggermente il parametro che adotta `TeX` per il calcolo dell'impaginazione delle righe. Axel Reichert ha postato in `de.comp.text.tex`¹¹ la sua personale modifica, grazie alla quale si evitano i principali problemi di sillabazione in un buon layout:

⁹ Cfr. anche [4, Sezione 2.7.2] e [6, Sezione 3.4]

¹⁰ <http://projekte.dante.de/DanteFAQ/Silbentrennung>

¹¹ Reperibile in Message-ID: <a84us0\$plqcm\$7@ID-30533.news.dfncis.de>

2 Comandi, classi e pacchetti obsoleti

Markus Kohm ha redatto uno script in Perl col quale si possono verificare on line i propri file per rintracciare gli errori più comuni (<http://kohm.de.tf/markus/texidate.html>).

Tuttavia bisogna notare che tale script non è un analizzatore (*parser*) TEX completo e che pertanto esso è in grado di rilevare solo gli errori più ovvi e diffusi. Prima provare, poi postare.

2.1 Comandi

2.1.1 Modifica dello stile del carattere

Nella Tabella 1 nella pagina seguente sono affiancati i vecchi e i nuovi comandi atti a modificare lo stile del carattere. Quelli vecchi non hanno mai funzionato in modo adeguato, anche perché il loro effetto quando sono dati in serie è piuttosto imprevedibile. Essi pertanto non vanno mai usati.

Ci sono validi sostituti che producono anche il corretto effetto cumulativo: ad esempio con `\sffamily\itshape` si sceglie un font senza grazie inclinato (cosa che non sarebbe avvenuta con `\sf\it`). Tuttavia è consigliabile usare la forma dichiarativa solo per lunghi tratti di testo all'interno di un ambiente; per tratti brevi o singole parole è meglio adottare i comandi con argomento (come `\textit`, `\textsf` ecc.), che inoltre liberano dal problema di chiudere i gruppi aperti. Si ricordi infatti che la sintassi corretta per `\sffamily` e simili è `{\sffamily ...}`, dove le graffe possono essere sostituite dai delimitatori di un ambiente. Le forme dichiarative sono comode solo in un caso: le celle di una tabella composta con `tabular`¹³.

Perché non si dovrebbero usare i comandi obsoleti? I comandi obsoleti non supportano le funzionalità del nuovo sistema di selezione dei font¹⁴ di $\text{L}\text{A}\text{T}\text{E}\text{X}_{2\epsilon}$. Ad esempio, `{\bf foo}` annulla tutte le impostazioni già fornite prima di stampare «foo» in nero. Ne consegue che non si può definire uno stile corsivo nero mediante `{\it \bf Test}` (questa definizione produce: **Test**).

I nuovi comandi `\textbf{\textit{Test}}` si comportano come atteso e producono: *Test*. Inoltre nei vecchi comandi non c'è nessuna «correzione» per il corsivo, ad esempio *rafforzare* (`{\it raf}forzare`) e *rafforzare* (`\textit{raf}forzare`).

2.1.2 Frazioni matematiche (`\over` vs. `\frac`)

Il comando `\over` non dovrebbe essere usato. `\over` è un comando di TEX , il quale, a causa della sintassi differente da quella di $\text{L}\text{A}\text{T}\text{E}\text{X}$, si può analizzare con più difficoltà o

¹³ Questo paragrafo è il precedente sono stati modificati dal traduttore.

¹⁴ NFSS: New Font Selection Scheme. Un panorama del NFSS è offerto da `fntguide` [5].

Tabella 1: Macro per modificare lo stile del carattere.

Obsolete	Sostituite in L ^A T _E X 2 _ε da	
	comandi	dichiarazioni
<code>{\rm ...}</code>	<code>\textrm{...}</code>	<code>\rmfamily</code>
<code>{\sf ...}</code>	<code>\textsf{...}</code>	<code>\sffamily</code>
<code>{\tt ...}</code>	<code>\texttt{...}</code>	<code>\ttfamily</code>
<code>{\it ...}</code>	<code>\textit{...}</code>	<code>\itshape</code>
<code>{\sc ...}</code>	<code>\textsc{...}</code>	<code>\scshape</code>
<code>{\sl ...}</code>	<code>\textsl{...}</code>	<code>\slshape</code>
<code>{\bf ...}</code>	<code>\textbf{...}</code>	<code>\bfseries</code>
—	<code>\textmd{...}</code>	<code>\mdseries</code>
—	<code>\textup{...}</code>	<code>\upshape</code>
—	<code>\emph{...}</code>	<code>\em^a</code>

^a Utile nella definizioni di macro. All'interno del testo non si dovrebbe usare `{\em ...}`, bensì `\emph{...}`.

non si può analizzare affatto. In particolare il pacchetto `amsmath` ridefinisce `\frac{ }{ }`, cosicché se si usa `\over` si riceverà un messaggio di errore. Un'altra ragione per usare `\frac{ }{ }` è che per l'utente – soprattutto nel caso di frazioni complesse – è più semplice comporre numeratore e denominatore.

▷ **NON:** `$a \over b$`

MA: `$$\frac{a}{b}$$`

2.1.3 Centrazione con `\centerline`

Il comando `\centerline` è ancora una volta un comando di T_EX e dovrebbe essere evitato in L^AT_EX. Da un lato `\centerline` è incompatibile con alcuni pacchetti di L^AT_EX (ad esempio `color`) e dall'altro il suo utilizzo può produrre effetti indesiderati o inattesi:

```
\begin{enumerate}
\item\centerline{Punto uno}
\end{enumerate}
```

Punto uno
1.

▷ **NON:** `\centerline{...}`

MA: `{\centering ...} oppure \begin{center} ... \end{center}`

Si noti tuttavia che la dichiarazione `\centering`, a differenza dell'ambiente definito da `\begin{center} ... \end{center}`, non agisce finché il capoverso non sia terminato esplicitamente mediante `\par` oppure con una riga vuota o ancora implicitamente mediante comandi successivi.

Nota Alcune osservazioni sulla centratura di oggetti grafici e di tabelle si leggono nella Sezione 3.1 a pagina 20.

2.2 Classi e pacchetti

2.2.1 scrlettr.cls vs. sclttr2.cls

La classe scrlettr.cls del pacchetto KOMA-Script è obsoleta e dev'essere sostituita dalla nuova sclttr2.cls. Per produrre un layout *simile* a quello della vecchia classe si può caricare l'opzione di classe KOMAold, che fornisce una modalità compatibile.

▷ **NON:** `\documentclass{scrlettr}`

MA: `\documentclass[KOMAold]{sclttr2}`

Nota Per nuovi testi e lettere si dovrebbe tuttavia adottare la nuova interfaccia, poiché è molto più flessibile.

Una spiegazione delle differenze fra le due interfacce utente esula dagli intenti di questo compendio, perciò rinviamo a scrguide [3].

2.2.2 epsf, psfig, epsfig vs. graphics, graphicx

I pacchetti epsf e psfig vanno sostituiti con graphics o con graphicx. epsfig è solo un wrapper¹⁵ per graphicx, la cui funzione è consentire l'utilizzo di graphicx per documenti vecchi che sono stati composti con psfig.

Dal momento che epsfig invoca graphicx, si *può* usare ancora, ma per documenti nuovi si dovrebbe ricorrere ai più recenti graphics o graphicx. La sintassi degli ultimi due pacchetti citati è estremamente flessibile. Il pacchetto epsfig è fornito essenzialmente per motivi di compatibilità.

Sulle differenze fra i pacchetti graphics e graphicx si veda grfguide [2]. Per una sintesi sulla centratura di oggetti grafici cfr. la Sezione 3.1 a pagina 20.

▷ **NON:** `\usepackage{psfig}`
`\psfig{file=figura,...}`

MA: `\usepackage{graphicx}`
`\includegraphics[...]{figura}`

2.2.3 doublespace vs. setspace

Per modificare l'interlinea si dovrebbe usare il pacchetto setspace. doublespace è obsoleto e dovrebbe essere rimpiazzato appunto con setspace. Vedere con attenzione anche la Sezione 1.4 a pagina 6.

▷ **NON:** `\usepackage{doublespace}`

MA: `\usepackage{setspace}`

¹⁵ Qui: un file di stile che ne carica un altro o alcuni altri.

2.2.4 fancyheadings, scrpage vs. fancyhdr, scrpage2

Il pacchetto fancyheadings è stato sostituito da fancyhdr. Un altro modo per personalizzare le testatine è usare il pacchetto scrpage2 di KOMA-Script. Anche in questo caso bisogna prestare attenzione a non usare scrpage. La documentazione di scrpage2 si trova in scrguide [3].

- | | | |
|---|---|---|
| ▷ | NON: <code>\usepackage{fancyheadings}</code> | MA: <code>\usepackage{fancyhdr}</code> |
| ▷ | NON: <code>\usepackage{scrpage}</code> | MA: <code>\usepackage{scrpage2}</code> |

2.2.5 La famiglia di pacchetti caption

Il pacchetto caption2 non dovrebbe più essere usato, perché ne esiste una nuova versione (la 3.x). Si dovrebbe prestare attenzione a non usare una versione troppo vecchia del pacchetto. Per esserne sicuri, caricare il pacchetto nel seguente modo:

- | | | |
|---|---|--|
| ▷ | NON: <code>\usepackage{caption}</code> | MA: <code>\usepackage{caption}[2008/08/24]</code> |
|---|---|--|

Chi avesse utilizzato in precedenza caption2, dovrebbe assolutamente prendere visione della la nuova documentazione [11, Sezione C.2].

2.2.6 isolatin, umlaut vs. inputenc

Osservazioni generali In generale, ci sono quattro possibilità per inserire Umlaut e altri caratteri non ASCII:

1. $H\{\backslash"u\}lle$: Il vantaggio di questo modo di procedere è che funziona sempre e su tutti i sistemi.
Per contro, gli svantaggi sono che la crenatura¹⁶ risulta disturbata, che nei testi tedeschi questa è di gran lunga la modalità più complicata di digitare e che il codice si legge con notevole difficoltà.
Questa variante – in ragione del problema alla crenatura – dovrebbe *sempre* essere evitata.
2. L'immissione nella forma $H\backslash"ulle$ o anche $H\backslash\{u\}lle$ non presenta il problema di crenatura sopra menzionato e si può adottare tranquillamente su qualunque sistema.
Gli svantaggi di questo modo di procedere sono anche qui l'immissione complicata e la cattiva leggibilità del codice.
Questa variante è la più indicata nella definizione delle macro e nei file di stile, perché è indipendente dalla codifica e dai pacchetti caricati.

¹⁶ La spaziatura positiva o negativa fra le lettere in relazione alle combinazioni di segni.

3. Con (n)german oppure con l'opzione (n)german del pacchetto babel si possono produrre le Umlaut digitando (H"ulle). Il vantaggio è ancora una volta che questo metodo funziona in tutti i sistemi. Poiché babel o anche (n)german sono disponibili in tutte le installazioni di T_EX, non dovrebbe aversi nemmeno alcun problema di compatibilità.

Gli svantaggi sono anche qui l'immissione complicata e la cattiva leggibilità del codice.

Questa variante funziona relativamente bene nel corpo di testo, ma dovrebbe essere evitata nella definizione delle macro e nei preamboli.

4. L'inserimento diretto (Hülle). I vantaggi qui sono evidenti. Il codice è scritto e si legge in modo "normale".

Lo svantaggio è che bisogna suggerire a L^AT_EX quale codifica di immissione si sta usando e che potrebbero sorgere eventuali problemi nello scambio di file tra sistemi diversi. Questo non è in sé un problema per T_EX o per L^AT_EX, ma potrebbe causare dei *problemi di visualizzazione* agli editor in sistemi diversi. Ad esempio, € codificato in iso-8859-15 (latin9) potrebbe essere *visualizzato* come Ⓔ in un editor per Windows (CP1252).

Questa variante funziona ottimamente per testi lunghi. Dovrebbe tuttavia essere evitata nella definizione di macro e nei preamboli.

Riassumendo, possiamo pertanto dire che nelle macro, nei preamboli e nei file di stile dovrebbe essere adottata la forma H\ "ulle oppure H\ {u}lle e all'interno del testo o H"ulle oppure Hülle.

Codifica dell'input Per suggerire a L^AT_EX quale codifica impostare *non* si dovrebbe usare isolatin1 e nemmeno isolatin oppure umlaut. Questi pacchetti sono obsoleti e nemmeno disponibili in tutti i sistemi.

È corretto invece utilizzare il pacchetto inputenc con le seguenti opzioni:

latin1/latin9 per sistemi Unix (latin1 è disponibile anche in Microsoft Windows e Mac OS X);

ansinew per Microsoft Windows;

applemac per Mac¹⁷;

cp850 per OS/2;

utf8 per «moderni» sistemi Unix e in generale per sistemi (ed editor) che adottano Unicode.

▷ **NON:** `\usepackage{isolatin1}` **MA:** `\usepackage[latin1]{inputenc}`

¹⁷ Nell'utilizzo di OS X si dovrebbe allo stesso modo adottare latin1/latin9, perché questa codifica è più adatta di applemac allo scambio con utenti di altri sistemi operativi. Tuttavia si dovrebbe prestare grande attenzione alla codifica di input degli editor utilizzati.

- ▷ **NON:** `\usepackage{umlaut}` **MA:** `\usepackage[latin1]{inputenc}`

Nota Il metodo sopra illustrato per dichiarare la codifica di input si usa nel «classico» utilizzo di motori per \LaTeX come \pdf\LaTeX . I recenti sviluppi \Xe\LaTeX e \Lua\LaTeX elaborano file codificati in Unicode anche senza dichiarare la codifica. Entrambi gli ultimi motori citati possono essere impiegati produttivamente, ma richiedono di solito una riscrittura dei «vecchi» sorgenti \LaTeX in relazione alla scelta della lingua e dei tipi di carattere e non possiedono ancora le capacità di microtipografia nella misura in cui ne gode \pdf\LaTeX .

2.2.7 `t1enc` vs. `fontenc`

Il pacchetto `t1enc` è obsoleto e dovrebbe pertanto essere sostituito da `fontenc`.

- ▷ **NON:** `\usepackage{t1enc}` **MA:** `\usepackage[T1]{fontenc}`

2.2.8 `natdin.bst` vs. `dinat.bst`

L'autore di `dinat.bst` raccomanda di sostituirlo con `natdin.bst`, perché il primo non è più stato sviluppato di recente, mentre il secondo è stato enormemente migliorato.

- ▷ **NON:** `dinat.bst` **MA:** `natdin.bst`

Nota In versioni precedenti di questo documento abbiamo fatto la raccomandazione esattamente opposta, che si fondava sullo stato di sviluppo di allora. I miglioramenti a `natdin.bst` che sono stati prodotti in seguito giustificano ora la preferenza a favore di quest'ultimo pacchetto.

2.2.9 `glossary` vs. `glossaries`

Il pacchetto `glossary` è obsoleto; come successore è stato creato dalla stessa autrice `glossaries`. Dettagli sulla modifica si trovano in [12].

2.2.10 `Sistyle` e `SIunits` vs. `siunitx`

I pacchetti `Sistyle` und `SIunits`, utili alla composizione delle unità fisiche, non sono più stati sviluppati. Le funzionalità di entrambi i pacchetti sono unificate nel successore `siunitx`. In ragione della sua notevole flessibilità, `siunitx` può sostituire anche pacchetti come `unitsdef`, `units`, `fancyunits` e `fancynum`.

- ▷ **NON:** `\usepackage{SIunits}`
`\unit{1}{\metre}` **MA:** `\usepackage{siunitx}`
`\SI{1}{\metre}`

2.2.11 subfigure e subfig vs. subcaption

I pacchetti subfigure e subfig non sono più sviluppati e sono incompatibili con alcuni pacchetti attuali come hyperref e caption. Come sostituto si suggerisce subcaption.

▷ **NON:**

```
\usepackage{subfig}
\begin{figure}
\subfloat[][testo]{figura}
\end{figure}
```

MA:

```
\usepackage{caption}
\usepackage{subcaption}
\begin{figure}
\begin{subfigure}{larghezza}
figura\caption{testo}
\end{subfigure}
\end{figure}
```

2.3 Caratteri

Il tema «Caratteri e \LaTeX » è fonte di «gioia» perenne in `de.comp.text.tex`, e in generale si avvia con la domanda: «perché i caratteri appaiono così frastagliati in Acrobat Reader?». Le più frequenti ma *sbagliate* risposte a questa domanda raccomandano di risolvere il problema ricorrendo a times oppure a pslatex. Il fatto è che con l'utilizzo di questi pacchetti non solo si impostano caratteri completamente diversi, ma pure – dal momento che questi pacchetti sono obsoleti – sorgono nuovi problemi. Se, come suggerito nella sezione 2.3.9, si utilizza Latin Modern come sostituto (quasi) identico per il carattere «standard» di \LaTeX (cioè Computer Modern), allora il problema dell'aspetto frastagliato dei caratteri in Adobe Reader non si presenta.

La fntguide [5] fornisce informazioni generali sul New Font Selection Scheme (NFSS) di $\LaTeX 2\epsilon$.

2.3.1 times

Il pacchetto times è obsoleto (si veda psnfss2e [9]). Esso imposta `\rmdefault` per il carattere Times, `\sfdefault` per Helvetica e `\ttdefault` per Courier *senza* tuttavia impostare i corrispondenti caratteri matematici. Inoltre l'Helvetica non è correttamente scalato e appare troppo grande. Se si volesse adottare la combinazione Times/Helvetica/Courier, si dovrebbe allora fare così:

▷ **NON:**

```
\usepackage{times}
```

MA:

```
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
```

Nota Il fattore di scalatura per helvet in combinazione con Times dovrebbe essere compreso fra 0.90 e 0.92.

2.3.2 mathptm

Il pacchetto mathptm è il predecessore di mathptmx e dunque non va più usato.

- ▷ **NON:** `\usepackage{mathptm}` **MA:** `\usepackage{mathptmx}`

2.3.3 pslatex

Il pacchetto pslatex include al suo interno mathptm + helvet (scalato) e tuttavia adotta un Courier scalato in modo eccessivo. Il principale svantaggio di pslatex è che *non* funziona con le codifiche T1 e TS1.

- ▷ **NON:** `\usepackage{pslatex}` **MA:** `\usepackage{mathptmx}`
`\usepackage[scaled=.92]{helvet}`
`\usepackage{courier}`

Nota su tutte le combinazioni Times/Helvetica Ci si può anche limitare all'uso di cmtt per un effetto da «macchina da scrivere», col che non serve più caricare courier.

2.3.4 palatino

Il pacchetto palatino si comporta come times (a parte ovviamente l'impostazione di `\rmdefault` a Palatino) e non dovrebbe pertanto essere più utilizzato.

- ▷ **NON:** `\usepackage{palatino}` **MA:** `\usepackage{mathpazo}`
`\usepackage[scaled=.95]{helvet}`
`\usepackage{courier}`

Nota Il fattore di scalatura per helvet in combinazione col carattere Palatino dovrebbe essere impostato a 0.95.

In realtà, nonostante l'Helvetica *non* sia il carattere migliore da combinare con il Palatino, anche se ne è stato a lungo il complemento più adatto liberamente disponibile e gratuito, la sostituzione sopra suggerita assomiglia per quanto possibile al «vecchio» accostamento. Nel frattempo sono stati resi disponibili i Vera/Bera Sans – un'alternativa a nostro avviso migliore.

Chi possieda un CD di CorelDraw® (anche vecchio¹⁸) può combinare molto felicemente il Palatino con i caratteri Frutiger¹⁹ oppure Optima²⁰. Nella sua homepage²¹ Walter Schmidt ha pubblicato i corrispondenti adattamenti (pacchetti bfr e bop) per usare questi caratteri con T_EX.

¹⁸ I caratteri che sono forniti con le versioni più vecchie di CorelDraw non contengono ancora alcun carattere per l'Euro!

¹⁹ Bitstream Humanist 777, pacchetto bfr per T_EX creato da Walter Schmid.

²⁰ Bitstream Zapf Humanist, pacchetto bop per T_EX creato da Walter Schmid.

²¹ Schriften für T_EX: <http://home.vr-web.de/was/fonts>

2.3.5 mathpple

Questo pacchetto è il predecessore di mathpazo. Manca di alcuni caratteri, che vengono allora mutuati dai font Euler, altri caratteri non si adattano bene al Palatino e alcune metriche del carattere non sono corrette. Per più ampie informazioni cfr. psnfss2e [9].

- ▷ **NON:** `\usepackage{mathpple}` **MA:** `\usepackage{mathpazo}`

2.3.6 utopia

Il pacchetto utopia non dovrebbe più essere utilizzato perché il suo successore fourier presenta un'interfaccia migliore. Per informazioni più precise cfr. psnfss2e [9].

- ▷ **NON:** `\usepackage{utopia}` **MA:** `\usepackage{fourier}`

2.3.7 Lettere greche matematiche non inclinate

I passaggi che seguono, segnati in rosso, non vanno intesi nel senso di «non si deve più fare così», ma che ora il pacchetto upgreek offre un modo più facile per immettere i caratteri greci. Per consigli sull'utilizzo si consulti sempre la documentazione upgreek [8].

Il trucco pifont

- ▷ **NON:** `\usepackage{pifont}`
`\newcommand{\uppi}{%`
`\Pisymbol{psy}{112}}`
`...`
`\uppi`
 oppure
`\newcommand[1]{\upgreek}{%`
`\usefont{U}{psy}{m}{n}#1}`
`...`
`\upgreek{p}`
- MA:** `\usepackage{upgreek}`
`...`
`$_uppi$`

Il trucco di babel

- ▷ **NON:** `\usepackage[greek,...]{babel}`
`\newcommand{\upgreek}[1]{%`
`\foreignlanguage{greek}{#1}}`
`...`
`\upgreek{p}`
- MA:** `\usepackage{upgreek}`
`...`
`$_uppi$`

2.3.8 euler vs. eulervm

Il pacchetto `euler` dovrebbe essere sostituito con `eulervm`, perché il primo presenta dei problemi di compatibilità con altri pacchetti e manca di alcuni miglioramenti nei dettagli:

- `\hbar` (`\hs\ash` in questo carattere) funziona ora in modo migliore;
- sono disponibili caratteri matematici in nero che includono simboli greci.

Per informazioni più precise cfr. `eulervm` [7].

▷ **NON:** `\usepackage{euler}`

MA: `\usepackage{eulervm}`

2.3.9 ae, aecompl e zefonts

I pacchetti `ae` (insieme a `aecompl`) e `zefonts` servono da soluzione di ripiego per un problema che riguarda le Umlaut e altre lettere accentate e che sorge quando si utilizzano i caratteri standard di \LaTeX , e cioè i Computer Modern (CM). Poiché questi esistono solo nella codifica `OT1`, le lettere accentate e i caratteri particolari delle lingue diverse dall'inglese, inesistenti in quella codifica, vengono "costruiti" con i segni corrispondenti (una *ü*, cioè, risulta dalla sovrapposizione di `''` a `'u'` e così via). Questi caratteri composti confondono l'algoritmo di sillabazione di \LaTeX , il che fa sì che ad esempio le parole con l'Umlaut non riescano a essere sillabate. I pacchetti `ae` e `zefonts` risolvono questo problema con l'utilizzo dei cosiddetti font virtuali.

Un altro problema, a questo connesso, non viene invece risolto con i pacchetti citati: se si produce un file PDF, in quest'ultimo i caratteri composti vengono visualizzati in modo (apparentemente) corretto, ma le parole con l'Umlaut non sono trattate correttamente nelle ricerche all'interno del file o se vengono copiate e incollate.

I problemi qui descritti non si presentano se si utilizza un carattere con codifica `T1` e in formato `Type 1`²². Perciò di recente sono stati progettate le famiglie di caratteri `cm-super` e `Latin Modern`. Poiché quest'ultima famiglia si comporta in modo più solido degli originali Computer Modern, raccomandiamo di sostituire i CM con `Latin Modern`.

▷ **NON:** `\usepackage{ae}`
oppure
`\usepackage{zefonts}`

MA: `\usepackage[T1]{fontenc}`
`\usepackage{lmodern}`

²² Si deve notare che `T1` e `Type 1`, nonostante si chiamino in modo simile, non hanno affatto lo stesso significato: `T1` descrive la codifica del carattere dal versante di \LaTeX , mentre `Type 1` è una dichiarazione su come i caratteri vengono rappresentati nel file PDF. Perciò l'utilizzo di un carattere nella codifica `T1` risolve il problema della sillabazione delle parole con l'Umlaut, ma non necessariamente quello della ricerca delle Umlaut in un file PDF.

3 Miscellanea

Questa sezione – con l’eccezione di 3.2 – presenta trucchi e consigli più generali di quelli compresi sotto il titolo di «peccati».

3.1 Oggetti flottanti – «figure», «table»

Per centrare un oggetto flottante si dovrebbe utilizzare `\centering` in luogo dell’ambiente `center`, poiché questo introduce fra l’oggetto stesso e il testo circostante uno spazio verticale aggiuntivo che nella maggior parte dei casi può risultare indesiderato.

▷ **NON:**

```
\begin{figure}
  \begin{center}
    \includegraphics{bild}
  \end{center}
\end{figure}
```

MA:

```
\begin{figure}
  \centering
  \includegraphics{bild}
\end{figure}
```

Nota Se tuttavia si tratta di centrare una regione all’interno del corpo del testo o dell’ambiente `titlepage`, allora questo spazio aggiuntivo può essere naturalmente voluto.

3.2 L’appendice

L’appendice viene introdotta dal *comando* `\appendix`. *Non si tratta* di un ambiente.

▷ **NON:**

```
\begin{appendix}
  \section{Blub}
\end{appendix}
```

MA:

```
\appendix
\section{Blub}
```

3.3 Formule matematiche

In generale, per la tipografia matematica avanzata si deve usare `amsmath`, eventualmente integrato con `mathtools`. Esso presenta nuovi ambienti, che sostituiscono innanzitutto `eqnarray`. Vantaggi del pacchetto:

- gli spazi all’interno e intorno agli ambienti sono consistenti;
- la numerazione delle equazioni è posizionata in modo da non sovrapporsi mai all’espressione matematica;
- i nuovi ambienti (ad esempio `split`) consentono di spezzare facilmente lunghe equazioni;
- possono essere definiti con facilità nuovi operatori (analogamente a `\sin` ecc.) con spaziature più adatte.

$$\begin{array}{l} a = b \\ b = c \\ a = c \end{array}$$

Figura 3: Esempio per `eqnarray*`

$$\begin{array}{l} a = b \\ b = c \\ a = c \end{array}$$

Figura 4: Esempio per `align*`

Avvertimento Se si utilizza il pacchetto `amsmath` non si dovrebbe *mai* fare uso degli ambienti `displaymath`, `eqnarray` e `eqnarray*`, in quanto essi sono incompatibili con `amsmath` e così si genererebbero ancora una volta spaziature inconsistenti (cfr. le Figure da 3 a 4 in questa pagina).

Invece l'ambiente `\[...]` è correttamente adeguato da `amsmath` e può essere usato in luogo di `displaymath`. Gli ambienti `eqnarray` e `eqnarray*` possono essere sostituiti con una prima approssimazione rispettivamente da `align` e `align*`. Per una trattazione completa delle possibilità di `amsmath` si consulti la documentazione `amslatex` [1].

▷ **NON:**

```
\begin{eqnarray}
a &=& b\\
b &=& c\\
a &=& c
\end{eqnarray}
```

MA:

```
\begin{align}
a &= b\\
b &= c\\
a &= c
\end{align}
```

3.4 L'uso di `\graphicspath`

Bisogna²³ premettere che, per motivi di sicurezza relativi soprattutto ma non esclusivamente a Windows, le più recenti versioni di \TeX (sia `MiKTeX` sia `TeX Live` dalla versione 2009) accettano una versione ridotta e protetta dell'esecuzione del comando `\write18` (che equivale a impartire una qualche istruzione: l'argomento di `\write18` è cioè interpretato come se fosse una linea di comando). \TeX consente di eseguire pienamente un certo numero di istruzioni (che riguardano fra gli altri `BibTeX`, `epstopdf`, ecc.), ne esegue altre in modo ristretto e infine non ne esegue affatto altre (come la cancellazione di file), che devono essere autorizzate dall'utente²⁴.

Di conseguenza i programmi non possono più usare percorsi assoluti (che cioè partono dalla radice del disco fisso), ma solo percorsi relativi alla directory dove risiede il file principale del documento.

²³ Questa sezione è stata interamente riscritta dal traduttore.

²⁴ A meno di non eliminare tutte le limitazioni allo stesso `\write18` indicandolo espressamente nella riga di comando che l'editor invia al sistema operativo per lanciare comandi come `pdflatex:opzione -shell-escape` di `TeX Live` e opzione `-enable-write18` di `MiKTeX`. Tuttavia in questo caso è necessario essere consapevoli dei rischi che tale operazione comporta per la sicurezza dell'intero sistema.

La diffusa macro `\graphicspath`, la cui comodità risiede nel fatto che, una volta impostata, evita di dover specificare ogni volta la/le directory in cui si trovano le immagini, dovrebbe essere evitata perché presenta quanto meno dei limiti:

1. la ricerca da parte di \TeX del file grafico da elaborare dura più a lungo rispetto a quando tale ricerca è affidata alla libreria `kpathsea` a cui ricorre normalmente \TeX per la ricerca dei file, anche se con gli attuali processori l'aumento di tempo è trascurabile;
2. i nomi dei file specificati senza percorso devono essere accuratamente distinti in Windows, che non fa distinzione fra lettere minuscole e maiuscole; questa distinzione deve avere luogo anche se eventuali file omonimi si trovano in subdirectory diverse.

Tali problemi non si manifestano se tutti i file grafici si trovano in una sola subdirectory di quella dove si trova il file principale del documento: perciò si può continuare a usare `\graphicspath` in tutta sicurezza. Se invece un documento complesso avesse le sue parti (ad esempio i capitoli) collocate nelle proprie rispettive subdirectory e se ogni parte conservasse a sua volta le proprie immagini in una subsubdirectory, la situazione si presenterebbe molto confusa e inutilmente complicata rispetto a quello che si potrebbe avere includendo ogni file grafico specificandone il nome completo di percorso e di estensione²⁵.

Il motivo per cui la ricerca del file grafico dura più a lungo usando `\graphicspath` discende dal fatto che \TeX conserva al suo interno una lista di percorsi specificati con questo comando. In un documento complesso, con molti capitoli che richiedano a loro volta percorsi diversi per le loro figure, la ricerca avviene scandendo ricorsivamente tutti i percorsi contenuti in quella lista e ricercando in ogni cartella specificata da ciascuno di questi percorsi. Invece se si specifica ogni file grafico con il suo percorso \TeX non deve ricercare nulla, ma punta al file specifico in quanto il nome completo del file è univoco.

Eguale problematico appare il ricorso alla variabile di ambiente `TEXINPUTS`²⁶. Questa variabile può essere impostata nelle modalità proprie dei vari sistemi operativi. Benché si tratti più di un'aggiunta che di una modifica vera e propria del percorso di sistema, essa è comunque permanente e pertanto andrà modificata per ogni nuovo documento (oltre al fatto che in vecchie versioni di Windows non esista la possibilità di avere distintamente una variabile `TEXINPUTS` di sistema e una di utente).

²⁵ Prima del 2009 veniva consigliato di non specificare l'estensione, perché `latex` e `pdflatex` potevano includere file grafici di formato mutuamente esclusivo; quindi era possibile che nella stessa cartella una certa immagine fosse presente sia in formato `.eps` sia in formato `.pdf`. Anche senza specificare l'estensione e senza modificare il file sorgente, `latex` avrebbe incorporato il file `.eps` mentre `pdflatex` il file `.pdf`. Poiché attualmente anche `pdflatex` può incorporare file in formato `.eps`, questo modo di procedere non ha più ragione d'essere, anzi può risultare dannoso

²⁶ Cfr. Risposta di David Carlisle al «Bug-Report» di Markus Kohm: <http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2618>

Il metodo maggiormente sicuro ed efficiente (valido anche per `\graphicspath`) consiste allora nel definire dei link simbolici mediante i classici comandi `ln -s` per i sistemi UNIX e `mklink` in Windows Vista o in Windows 7 (questo procedimento è irrealizzabile nelle precedenti versioni di Windows, che non conoscono i link simbolici). A differenza di quanto accadeva con `TEXINPUTS`, tali link, in quanto locali, non apportano nessuna modifica ai percorsi di sistema. Grazie a essi nella/e directory del documento risultano (virtualmente) presenti, dovunque siano fisicamente collocate all'interno del disco fisso, le directory delle immagini, che dunque sono individuabili da `\write18` parzialmente abilitato.

È sempre opportuno consultare la documentazione del sistema operativo e della distribuzione $\text{T}_{\text{E}}\text{X}$ utilizzati.

3.5 Le macro `\...name`

Poiché in `de.comp.text.tex` si domanda di tanto in tanto come si possa modificare a esempio «Bibliografia» in «Elenco delle fonti», nella Tabella 2 nella pagina seguente sono indicate le macro (italiane) corrispondenti. Più precisamente, esse sono tratte dai file `italian.ldf` per `babel` e `gloss-italian.ldf` per `polyglossia`.

Ad esempio, per rinominare l'«Elenco delle figure» come «Figure», si adotti il comando seguente:

```
\renewcommand*{\listfigurename}{Figure}
```

Le altre macro si possono rinominare in modo analogo. Se si adotta `babel` è opportuno operare con `\addto`. Infatti a ogni reimpostazione della lingua (nel nostro caso l'italiano) viene reiterato il comando `\captionsitalian`: la personalizzazione andrebbe così persa e reintrodotta il termine di default. Il comando `\addto` è definito in `babel`, ma si può applicare anche a `polyglossia` prestando però attenzione a caricare in aggiunta `etoolbox` o simili. Cfr. anche le *De-TeX-FAQ* su questo tema²⁷.

La scrittura corretta è pertanto:

```
\addto{\captionsitalian}{%
  \renewcommand*{\listfigurename}{Figure}}
```

A chi adotta *KOMA-Script* raccomandiamo di usare l'equivalente comando `\renewcaptionname`:

```
\renewcaptionname{italian}{\listfigurename}{Figure}
```

Questo comando dev'essere dato dopo `\begin{document}`.

²⁷ «I miei tentativi di ridefinire `\figurename`, `\tablename` e simili con l'utilizzo del pacchetto `babel` falliscono. Che cosa posso fare?» in <http://projekte.dante.de/DanteFAQ/Verschiedenes>

Tabella 2: *Le macro definite da babel e polyglossia con l'opzione italian*

Nome della macro	Definizione originale	Resa italiana
<code>\prefacename</code>	Preface	Prefazione
<code>\refname^a</code>	References	Riferimenti bibliografici
<code>\abstractname</code>	Abstract	Sommario
<code>\bibname^b</code>	Bibliography	Bibliografia
<code>\chaptername</code>	Chapter	Capitolo
<code>\appendixname</code>	Appendix	Appendice
<code>\contentsname</code>	Contents	Indice
<code>\listfigurename</code>	List of Figures	Elenco delle figure
<code>\listtablename</code>	List of Tables	Elenco delle tabelle
<code>\indexname</code>	Index	Indice analitico
<code>\figurename</code>	Figure	Figura
<code>\tablename</code>	Table	Tabella
<code>\partname</code>	Part	Parte
<code>\enclname</code>	encl	Allegati
<code>\ccname</code>	cc	e p. c.
<code>\headtoname</code>	To	Per
<code>\pagename</code>	Page	Pag.
<code>\seename</code>	see	vedi
<code>\alsoname</code>	see also	vedi anche
<code>\proofname</code>	Proof	Dimostrazione
<code>\glossaryname</code>	Glossary	Glossario

^a Solo nella classe article.

^b Solo nelle classi report e book.

A Esempio di \sloppy

Questo è il codice di esempio che Markus Kohm ha pubblicato in [de.comp.text.tex](#):

```
\documentclass{article}

\setlength{\textwidth}{20em}
\setlength{\parindent}{0pt}
\begin{document}
\typeout{First without \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\ -ta\ -tata
tatata tatata tatata tatata tatata tatata tata\ -tata
tatata tatata tatata tatata ta\ -tatatatatt\ -ta
tatata tatata tatata tatata tatata tatata ta\ -ta\ -ta\ -ta

\typeout{done.}

\sloppy
\typeout{Second with \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\ -ta\ -tata
tatata tatata tatata tatata tatata tatata tata\ -tata
tatata tatata tatata tatata ta\ -tatatatatt\ -ta
tatata tatata tatata tatata tatata tatata ta\ -ta\ -ta\ -ta

\typeout{done.}
\end{document}
```

Fonte: [Message-ID: <8557097.gEimXdBtjU@ID-107054.user.dfncis.de>](#)

Riferimenti bibliografici

- [1] American Mathematical Society, *User's Guide for the amsmath Package*, versione 2.0, febbraio 2002 (consultabile con `texdoc amsmath`).
URL: [CTAN://macros/latex/required/amslatex/](http://ctan.org/macros/latex/required/amslatex/)
- [2] Carlisle, David P. e The L^AT_EX₃ Project, *Packages in the 'graphics' bundle*, novembre 2005 (consultabile con `texdoc graphicx`).
URL: [CTAN://macros/latex/required/graphics/](http://ctan.org/macros/latex/required/graphics/)
- [3] Kohm, Markus e Morawski, Jens-Uwe, *KOMA-Script, a versatile L^AT_EX_{2 ϵ} bundle*, versione 3, maggio 2011 (consultabile con `texdoc scrguien`).
URL: [CTAN://macros/latex/supported/koma-script/](http://ctan.org/macros/latex/supported/koma-script/)
- [4] The L^AT_EX₃ Project, *L^AT_EX_{2 ϵ} for class and package writers*, febbraio 2006 (consultabile con `texdoc clsguide`).
URL: [CTAN://macros/latex/doc/clsguide.pdf](http://ctan.org/macros/latex/doc/clsguide.pdf)
- [5] L^AT_EX₃ Project Team, *L^AT_EX_{2 ϵ} font selection*, novembre 2005 (consultabile con `texdoc fntguide`).
URL: [CTAN://macros/latex/doc/fntguide.pdf](http://ctan.org/macros/latex/doc/fntguide.pdf)
- [6] L^AT_EX₃ Project Team, *L^AT_EX_{2 ϵ} for authors*, novembre 2005 (consultabile con `texdoc usrguide`).
URL: [CTAN://macros/latex/doc/usrguide.pdf](http://ctan.org/macros/latex/doc/usrguide.pdf)
- [7] Schmidt, Walter, *The Euler Virtual Math Fonts for use with L^AT_EX*, versione 4, gennaio 2005 (consultabile con `texdoc eulervm`).
URL: [CTAN://fonts/eulervm/](http://ctan.org/fonts/eulervm/)
- [8] Schmidt, Walter, *The upgreek package for L^AT_EX_{2 ϵ}* , versione 2.0, febbraio 2003 (consultabile con `texdoc upgreek`).
URL: [CTAN://macros/latex/contrib/upgreek/](http://ctan.org/macros/latex/contrib/upgreek/)
- [9] Schmidt, Walter, *Using common PostScript fonts with L^AT_EX*, `PSNFSS` versione 9.2, settembre 2004 (consultabile con `texdoc psnfss2e`).
URL: [CTAN://macros/latex/required/psnfss/psnfss2e.pdf](http://ctan.org/macros/latex/required/psnfss/psnfss2e.pdf)
- [10] Schmidt, Walter, Knappen, Jörg, Partl, Hubert e Hyna, Irene, *L^AT_EX_{2 ϵ} -Kurzbeschreibung*, versione 2.3, aprile 2003 (consultabile con `texdoc l2kurz2`).
URL: [CTAN://info/lshort/german/](http://ctan.org/info/lshort/german/)
- [11] Sommerfeldt, Axel, *Anpassen der Abbildungs- und Tabellenbeschriftungen mit Hilfe des caption-Paketes*, versione 3.2e, novembre 2011 (consultabile con `texdoc caption-deu`).

Riferimenti bibliografici

URL: [CTAN://macros/latex/contrib/caption/](http://ctan.org/macros/latex/contrib/caption/)

Disponibile allo stesso indirizzo anche nella versione inglese *Customizing captions of floating environments using the caption package*.

- [12] Talbot, Nicola L. C., *Upgrading from the glossary package to the glossaries package*, aprile 2011 (consultabile con texdoc `glossary2glossaries`).

URL: [CTAN://macros/latex/contrib/glossaries/glossary2glossaries.pdf](http://ctan.org/macros/latex/contrib/glossaries/glossary2glossaries.pdf)