

Recently there were some queries in  $\TeX$ hax and UK $\TeX$  about a  $\TeX$  overflow while using the `doc` option. The source of this is the size of  $\TeX$ 's save stack: usually Don Knuth's original value (600) is used. This works well with plain  $\TeX$ , but it is much too small for  $\LaTeX$  documents.<sup>2</sup> Even size changes in the argument of a `\caption` command can result in an overflow of the save stack! Therefore we strongly suggest to all  $\TeX$  implementors that they increase this parameter to a value at least as high as 1500.

## References

- [1] Frank Mittelbach and Rainer Schöpf, "With  $\LaTeX$  into the Nineties." Talk given at the 10th Anniversary TUG meeting, Stanford, August 1989, *TUGboat* 10#4, to appear.
- [2] Rainer Schöpf and Frank Mittelbach, " $\LaTeX$  limitations and how to get around them." Talk given at the Euro $\TeX$  conference, Karlsruhe, September 1989, to appear.

◊ Frank Mittelbach  
Electronic Data Systems  
(Deutschland) GmbH  
Eisenstraße 56  
D-6090 Rüsselsheim  
Federal Republic of Germany  
Bitnet: pzf5hz@drueds2

◊ Rainer Schöpf  
Institut für Theoretische Physik  
der Universität Heidelberg  
Philosophenweg 16  
D-6900 Heidelberg  
Federal Republic of Germany  
Bitnet: BK4@DHDURZ1

<sup>2</sup> A similar problem arose with  $\TeX$ 's main memory size and hash table size which were increased when  $\LaTeX$  was released.

---

## The Development of National $\LaTeX$ Styles

Johannes Braams, Victor Eijkhout, and  
Nico Poppelier

### Abstract

At its autumn 1988 meeting, the Dutch  $\TeX$  users group (NTG) established a working group (number 13) that was to concentrate on the problems involved in the use of  $\TeX$  for Dutch texts. Since then the working group, which includes the authors, has created a number of style options for  $\LaTeX$  that remedy some common problems with the non-English use of  $\LaTeX$ , and is along the way developing document styles that are compatible with the standard styles, but have a layout that is more palatable for Dutch users. In this article we treat implementation aspects of the styles and style options, and we discuss some matters of layout.

### 1 The need for national $\TeX$

On several occasions it is stressed in both *The  $\TeX$ book* [1] and the  $\LaTeX$  book [2] that non-English users of  $\TeX$  may have to take steps in order to adapt  $\TeX$  to their native language. For several languages such steps have indeed been taken, for instance for the German language [3]. It was only natural that the Dutch  $\TeX$  users group (NTG) would also initiate an effort in this direction. Thus the active life of working group 13 began somewhere about the beginning of 1989.

As the use of  $\LaTeX$  is quite wide-spread in the Netherlands, and because most matters of national standardization can be conveniently handled in the context of document styles — and also because the Dutch language does not have the problems of national characters that are prominent for  $\TeX$  users to the North, South and East of this country — it was decided to focus mainly on the development of national styles and style options for  $\LaTeX$ . This article treats some of the problems encountered and the way they were solved.

### 2 The 'chapter' problem

One of the first problems non-English users of  $\LaTeX$  run into, is that of English terms ('Abstract', 'Contents') contained in the document styles. The resourceful user, or the  $\TeX$ nician consulted, will probably take out a text editor and hunt through the style file for the offending string, replacing it by its equivalent in his/her native language. This process will most likely result in new styles, called (for Dutch) `artikel`, `rapport` and so on.

At this point it may occur to the conscientious user that Leslie Lamport must have foreseen this situation, and probably have made provisions for it, so why not see if there is a suggested approach for this. A cursory perusal of the table of contents of the L<sup>A</sup>T<sub>E</sub>X book will then lead our user to section 5.1.4 ‘Customizing the document style’. There Lamport takes half a page to explain how the word ‘Chapter’ is really just the value of a control sequence `\@chapapp`. This implies that changing this text to ‘Hoofdstuk’ does not require editing of document styles at all, it merely needs a one-line option file. Charming, one would say. However, immediately after that there is a strange sentence ‘You may also want to redefine the `\appendix` command, replacing `Appendix [...]`’. Curiouser and curiouser! Can’t I just redefine some control sequence that yields the word ‘Appendix’? Well, as it turns out, ‘Chapter’ is the *only* text that has been parametrized, the rest is hard-wired into the document styles.

And such is the situation in which working group 13 found itself: we could think of at least three ways of solving the ‘Chapter’ problem.

- We could make exact copies of the standard styles and all point size options (`art10.sty` and such), replacing all English text by Dutch text. The disadvantages of this are that (1) supposing a certain installation of T<sub>E</sub>X will be used for three languages, then every style file has to be present three times, and (2) if Lamport then finds a bug or decides to issue upgrades of the standard styles, this would require a multitude of changes.
- We might also rewrite the standard styles, parametrizing them, and add option files for the different languages that contain only parameter settings. This solution is subject only to the second objection above.
- In fact there exists an even better solution (due to Piet van Oostrum): if an option file would be able to find out what style is being used, it can replace and parametrize just those commands that contain text, and afterwards set the parameter values to some appropriate language. This approach is probably the most economical one: the original styles are still used, and, if the option file contains parameter settings for a number of languages, this single option file suffices.

The option file `dutch`, created by one of the authors (JB), implements the third possibility above. It uses for parameter names those suggested by Herbert Partl [3] in the `german` style. Note however,

that `german` — an implementation of the second possibility above — does not do the actual redefinitions but merely sets the parameters. Thus it will only function correctly with edited standard styles.

As an example of parametrization of commands consider the following definition, taken from `article.sty`<sup>1</sup>

```
\def\abstract{\if@twocolumn
  \section*{Abstract}
\else \small
  \begin{center}
    {\bf Abstract\vspace{-.5em}}
  \end{center}
  \quotation
\fi}
```

This definition is overridden in `dutch.sty` by

```
\def\abstract{\if@twocolumn
  \section*{\abstractname}
\else \small
  \begin{center}
    {\bf \abstractname\vspace{-0.5em}}
  \end{center}
  \quotation
\fi}
```

to which is added a command initializing the language-dependent parameters:

```
\def\captionsdutch{
\def\abstractname{Samenvatting}
.
.
}
```

Some comments about this approach are in order. First of all, the option file `dutch.sty` really consists of two disparate parts, the redefinitions and the initializations. The redefinitions are only to make up for what we see as a deficiency in the distribution style files. The parameter initializations are then the truly language-dependent part.

Secondly, the option `dutch` obviously works with the standard document styles, but will collide with some other styles, for instance with new styles to be developed — as we found out to our chagrin. The reason for this is that the same mechanism that repairs the `article` style, will attempt to repair any style that looks like it. Thus, if a Dutch `artikel` style has its own ideas about how an abstract should look, it must have a way of protecting itself against `dutch`’s zeal. Such a possibility exists, and it has been incorporated in the new Dutch styles that will

<sup>1</sup> The macros presented here have been simplified to convey the essence of what we are telling. They are not usable in this form.

be discussed below. Other non-standard styles may need to be edited before they can be used with the `dutch` option, however.

### 3 Making the layout less ‘loud’

To Dutch—and probably some other—eyes, the L<sup>A</sup>T<sub>E</sub>X styles are a bit ‘loud’, and therefore two style options have been developed by one of the authors (NP). These work together with the standard styles and make the general layout a bit more compact. A first option is `a4.sty` which sets various parameters in order to accommodate European standard A4 paper. This option started out as a bare union of the `a4` style option of John Pavel and the `a4wide` style option of Jean-François Lamy, but the current version has undergone fine tuning. The second option is called `sober`: it reduces the sizes of fonts in section headings, and it eliminates the white spaces surrounding section headings and items in list structures.

### 4 Compatible replacement styles for Dutch

Without asserting that there exists such a thing as ‘a typically Dutch layout’, we can still state that certain aspects of the L<sup>A</sup>T<sub>E</sub>X styles are less desirable for Dutch documents. Therefore the working group at its first meeting already declared it a goal to develop styles with a Dutch look. With the guidance of a graphical designer [4], and using some books on the subject [5, 6, 7]—including one by a Dutch typographer—two styles have since then reached completion, implemented by one of the authors (VE). One style is compatible with `article`, and one with `report`. By compatibility we mean here that the new style implements the same commands as one of the original styles. Both styles are based on the same graphical design. It is our intention to have further compatible styles available in the near future based on different layouts.

#### 4.1 The flexibility of L<sup>A</sup>T<sub>E</sub>X

Probably the easiest way to develop a new document style is to start out with an already existing one, and to modify it gradually. In this process one discovers that L<sup>A</sup>T<sub>E</sub>X has a lot of possibilities for modification built into it. For instance, whether or not the sixth parameter of `\@startsection`—the generic command used in document styles to define section headings—is positive controls the placement of the section heading above or embedded in the text. The absolute value of this parameter is then either

- the vertical distance between section heading and the first line of the text when it is positive, or
- when negative it is the horizontal distance between the run-in heading and the first word of the text.

On the other hand a number of design decisions have been hard-wired into the sectioning commands, and cannot be easily changed, for instance, by means of the parameters of `\@startsection`. Consider as an example the distance between the section number and the heading. This turns out to be exactly ‘1 em’ in the font of the section heading, as can be seen from the following definition—again, this is a rather simplified form of the definition actually appearing in `latex.tex`.

```
\def\@sect#1#2#3#4#5#6[#7]#8{
  \refstepcounter{#1}
  % #1 is ‘subsection’ for example
  \edef\@svsec{\csname the#1\endcsname
    \hskip 1em }
  \begingroup#6\relax % #6 is the style
    \@hangfrom{\hskip #3\relax\@svsec}%
    {\interlinepenalty=\@M
     #8\par}
  % #8 is the heading text
  \endgroup \@xsect{#5}}
```

Clearly, L<sup>A</sup>T<sub>E</sub>X makes it easy for the user to determine whether the section number be set using digits or roman numerals or letters: the only action required is redefinition of `\thesection` and so on. Also it is easy for the document-style designer to specify the style of the section heading: this is determined by the sixth parameter of `\@startsection`. It is not easy to change that ‘1 em’, which one might just want to do occasionally.

#### 4.2 Going Dutch

The styles `artikel` and `rapport` embody a number of changes with respect to `article` and `report`. Some of these are fairly trivial, such as the fact that we switch on ‘french spacing’, and some are more complicated. In the rest of this section we will treat the most significant change, the notion of a ‘unit indent’, from both a typographical and implementer’s point of view.

One of the principles of document design is<sup>2</sup> that the eye is able to pick up regularities in a page layout, and that their presence is considered posi-

<sup>2</sup>Or rather: seems to be. We have not encountered explicit statements to this effect, but implicitly it seems to be there.

tive, but that having too much variation is confusing. For instance, the designer we consulted insisted that the white space separating a section heading and the following text should bear some simple relation to the `baselineskip`, and should not have any stretch.

One point that all our sources seemed to agree on was that the number of ‘implied left margins’ in a document should be as low as possible. By an implied left margin we mean here a non-zero distance from the actual left margin that is taken by more than one item of the document. Examples of implied left margins are

- the paragraph indentation; furthermore
- the left margins of items in an ‘itemize’ or ‘enumerate’ list construct, and
- the left (or right) sides of the numbers and labels in such list constructs, but also
- the left side of the text of a section heading.

In the standard styles of L<sup>A</sup>T<sub>E</sub>X all of these four distances are independent and are different from one another. In the style we have developed it was decided to strengthen the visual coherence of the layout by taking the same value for each of them whenever possible.

Implementing this idea meant adopting a new dimension `\unitindent` which first of all unifies the `\parindent` and the `\leftmargini`, the indentation of non-embedded lists.

```
\newdimen\unitindent
{\setbox0=\hbox{\normalsize\rm 2.2.2
                \hskip.5em}
 \global\unitindent=\wd0}
\parindent=\unitindent
\leftmargini=\unitindent
```

Admittedly this will give a rather large indentation, but this does not seem to be uncommon in contemporary typographical design. In fact, while the L<sup>A</sup>T<sub>E</sub>X book takes the classical quad — the Dutch term translates to ‘a square of white’ — for the paragraph indent, *The T<sub>E</sub>Xbook* shows a large indentation which is equal to that for lists on the outer level.

From the computation of the size of the unit indent, the reader may have gathered already that it will also be put to another use: we want the text of section headings — of all numbered sections — to appear at a distance of `\unitindent` from the left margin. As was indicated above, this requires some modification of the `\@sect` macro. We therefore include in the style file the following redefinition:

```
\def\@sect#1#2#3#4#5#6[#7]#8{
```

```
\refstepcounter{#1}
  % #1 is ‘section’ for example
\edef\@svsec{\hbox to \unitindent
  {\csname the#1\endcsname \hfil}}
\begingroup #6\relax
  % #6 is the style
  \@hangfrom{\hskip #3\relax\@svsec}%
  {\interlinepenalty=\@M
   \hyphenpenalty=\@M
   \exhyphenpenalty=\@M
   \rightskip=0cm plus 13cm
   #8\par}
  % #8 is the heading text
\endgroup \@xsect{#5}}
```

where the macro `\@svsec` for the heading now gives a horizontal box of a predetermined width. Note that we have also ensured that the heading is set ragged right. Hyphenation in a heading — even of words with an explicit hyphen — is something so hideous, that we suspect this to be an oversight of Leslie Lamport.

A further unification of implied left margins can be achieved if we look more carefully at embedded lists. In the standard styles an embedded list construct defines two left margins: the left margin of the items, and the left side of the labels. This second margin can be eliminated by making it equal to the text margin of the surrounding list. The placement of such labels is governed by the value of `\labelwidth` and by the macro `\makelabel` which is passed by `itemize` and `enumerate` to `\list`. We therefore specify that the label will take the full width of the indentation, for instance

```
\def\@listii{\leftmargin=\leftmarginii
 \labelwidth=\leftmarginii}
and redefine \itemize (in the document style file)
\def\itemize{\advance\@itemdepth \@ne
 \edef\@itemitem{\labelitem
 \romannumeral\the\@itemdepth}%
 \list{\csname\@itemitem\endcsname}%
 {\def\makelabel##1{##1\hfil}}}
```

so that it will make labels that are flush left. The result of these actions can be seen in figures 1, 2, and 3.

### 4.3 The international feel

It has occurred to us that L<sup>A</sup>T<sub>E</sub>X users outside the Netherlands may also like the new styles, and may also want to use them. We decided therefore to make the styles in a sense truly compatible to `article` and `report`: when used on their own the styles will produce English captions, and using them in combination with `dutch` or `german` will alter these. However,

Figure 1:

# 1 Comparison of styles

## 1.1 Properties of the distribution styles

There are people who criticize the layout of the  $\text{\LaTeX}$  distribution style files.

Main point of contention is usually a perceived lack of unity, in particular the fact that the indentations for

- lists, paragraph indentation, and the implied indentation of
- section labels on levels
  1. for sections,
  2. for subsections,
  3. and subsubsections

are all different.

Also the layout is sometimes considered 'loud': fonts in headings tend to be quite big, and there is a lot of white space in the layout. Maybe such things are regionally determined; for use in Dutch, in any case, it was necessary to change them.

Figure 2:

# 1 Comparison of styles

## 1.1 Properties of the distribution styles

There are people who criticize the layout of the  $\text{\LaTeX}$  distribution style files.

Main point of contention is usually a perceived lack of unity, in particular the fact that the indentations for

- lists, paragraph indentation, and the implied indentation of
- section labels on levels
  1. for sections,
  2. for subsections,
  3. and subsubsections

are all different.

Also the layout is sometimes considered 'loud': fonts in headings tend to be quite big, and there is a lot of white space in the layout. Maybe such things are regionally determined; for use in Dutch, in any case, it was necessary to change them.

Figure 3:

# 1 Comparison of styles

## 1.1 Properties of the distribution styles

There are people who criticize the layout of the  $\text{\LaTeX}$  distribution style files.

Main point of contention is usually a perceived lack of unity, in particular the fact that the indentations for

- lists, paragraph indentation, and the implied indentation of
- section labels on levels
  1. for sections,
  2. for subsections,
  3. and subsubsections

are all different.

Also the layout is sometimes considered 'loud': fonts in headings tend to be quite big, and there is a lot of white space in the layout. Maybe such things are regionally determined; for use in Dutch, in any case, it was necessary to change them.

the reader will understand after the above discussion that we achieve this by somewhat more sophisticated means than are employed in the standard styles.

None of the commands in the `artikel` style, for instance, contain actual texts. Instead they contain such commands as `\abstractname`. These commands are initialized at the end of the file to give English texts. Any language option based on the same parameter names, such as `dutch` or `german`, can override these settings for use in other languages. Naturally, the `artikel` style takes precautions to prevent `dutch` from mistaking it for `article`.

On a more philosophical note, we may add that we feel that this situation is how it should have been from the beginning. In fact, Leslie Lamport himself has shown the way to international styles — see the passage quoted above. For some reason however, he stopped short and did not implement this idea.<sup>3</sup>

## 5 Conclusion

It has turned out to be possible to adapt L<sup>A</sup>T<sub>E</sub>X for use with the Dutch language, in such a way that hardly any action on the part of the user is required. With some style options English terms can be replaced, and the typically American layout can be made more acceptable to Dutch eyes. The development of completely new document styles for Dutch, however, leaves the implementer with the feeling that he has had to unearth sections of L<sup>A</sup>T<sub>E</sub>X that were never meant to be rewritten.

## References

- [1] Donald Knuth, *The T<sub>E</sub>Xbook*, Addison-Wesley, 1986.
- [2] Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X, A Document Preparation System*, Addison-Wesley, 1986.
- [3] Hubert Partl, *German T<sub>E</sub>X*, *TUGboat*, vol 9 (1988), no 1, pp. 70–72.
- [4] Inge Eijkhout, *Conversations and correspondence*, 1989.
- [5] John Miles, *Ontwerpen voor Desktop Publishing, Alles over layout en typografie op de personal computer*, Uitgeverij Mingus, Baarn 1988, Dutch

translation of *Design for Desktop Publishing: a guide to layout and typography on the personal computer*, Fraser, London 1987.

- [6] Stanley Morison, *Grondbeginselen der Typografie*, Uitgeverij de Buitenkant, Amsterdam 1987, Dutch translation of *First Principles of Typography*.
- [7] Karel Treebus, *Tekstwijzer, een gids voor het grafisch verwerken van tekst*, SDU uitgeverij, The Hague 1988. (A book covering all aspects of typographical design. In Dutch.)

◇ Johannes Braams  
 PTT Research Neher Laboratories  
 St. Paulusstraat 4  
 2264 XZ Leidschendam  
 The Netherlands  
 all nets: JL.Braams@pttrnl.nl

◇ Victor Eijkhout  
 Department of Mathematics  
 University of Nijmegen  
 Toernooiveld 5  
 6525 ED Nijmegen  
 The Netherlands  
 Bitnet: U641000@HNYKUN11

◇ Nico Poppelier  
 University of Utrecht  
 Department of Physics  
 Postbus 80.000  
 3508 TA Utrecht  
 The Netherlands  
 Bitnet: Poppelier@HUTRUU51

<sup>3</sup> Editor's Note: At the TUG meeting, Leslie Lamport met with a group of users "seriously interested in the future of L<sup>A</sup>T<sub>E</sub>X" and established a committee, chaired by Frank Mittelbach, to undertake maintenance of the L<sup>A</sup>T<sub>E</sub>X code and creation of L<sup>A</sup>T<sub>E</sub>X version 2.10 (and 3.10) (see p. 400 in this issue of *TUGboat*). It is likely that "internationalization" of the L<sup>A</sup>T<sub>E</sub>X code will be one of the results of this committee's work.