`Manuscript`—A Package Emulating Typewriter "Typesetting"

Matěj Cepl, `matej at ceplovi dot cz`

This document describes package `manuscript`
version 1.7, from 2015/01/19

1 Purpose

Purpose of this package is to emulate appearance of the document written on classical typewriter as much as possible. So far, whenever backward requirement of some institutions (especially universities) to provide paper or thesis in the layout developed in times before personal computers emerges on any TEX-related public forum, it is met with strong (and sometimes even angry) resistance and many advise how to overcome resistance of the institution. I totally agree with the basic premise of this attempt (of course, TEX was created for making "masterpieces of typography" not to emulate typewriters).

Having said that, this package goes exactly in the opposite direction than these people advise. Instead of trying to avoid typewriter-driven layout, it tries to emulate it as much as possible. Of course, the most important motivation for such package is challenge and curiosity how far I can get using just TEX tools in this endeavor. However, there are also some real reasons why this package might be useful. First of all there are situations when the directives of backward layout are non-negotiable (e. g., in commercial setting or with too stiff university). Moreover, my conciliatory character leads me more to honor other cultures (and although very short-lived and feeble, there *was* a typographical culture of typewriters) rather then rejecting them. Actually, during work on this package my appreciation of strict puritanical simplicity of typewriters just grew (and of course, it is obvious that in some aspects typesetting documents with this package would create documents of the quality never possible with a real typewriter—just TEX's optimal line breaking with few divided words makes a lot of difference).

If you like it, enjoy! If not, sorry, just use another package.

Another objective was to secure compatibility both with classical `article`-like packages as well as with packages from `Koma-script` family.

2 The Coding

`MS@ps`

The first of all we need to create new condition `MS@ps` to control package options—whether font Courier should be used (if true) or `cmtt`. We cannot use `ifthen` package as it redefines catcodes and conflicts with the redefinition of quotes for `cmtt` font (see below).

```
\newif\ifMS@ps
```

`\DeclareOption`
`\ExecuteOptions`
`\ProcessOptions`

Declare the options by setting `MS@ps` variable. The options `cm` and `cr` set use of font `cmtt` or Courier, respectively. We want to use Courier as default option.

```
\DeclareOption{cm}{\MS@psfalse}
\DeclareOption{cr}{\MS@pstrue}
\ExecuteOptions{cr}
\ProcessOptions\relax
```

We need to read some additional packages which are needed for good working of the package: `setspace` because whole document should be doublespaced (except for footnotes etc.), `fontenc` because we should be able to print all European characters (it could be redefined in the document itself *after* loading `manuscript` package, and `ragged2e` because of linebreaking and ragged justification.

```
\RequirePackage{setspace}
\RequirePackage[T1]{fontenc}
\RequirePackage[NewCommands]{ragged2e}
\RequirePackage{soul}
```

`\rmdefault`

We need to set up the default font for Roman characters based on the boolean variable `MS@ps`.

```
\ifMS@ps
   \renewcommand{\rmdefault}{pcr}
\else
   \renewcommand{\rmdefault}{cmtt}
\fi
```

`\descfont`
`\sectfont`

There is no bold in `cmtt`, so that I redefine also some fonts—usefull only for koma-script package, because I do not care too much for `article :-)`.

```
\@ifundefined{scr@parskip}{}{%
   \renewcommand*{\descfont}{\scshape}
   \renewcommand*{\sectfont}{\large\scshape} }
```

`\MS@q*`
`‘‘`
`’`

This is the most obscure part of the package.[1] There are no problems with `Courier` font, but `cmtt` does not have some characters needed for normal writing—especially quotes. We have to change catcode of , and ‘ characters and redefine couples of these characters to be printed in `cmss` font in the Old Knuth’s coding OT1.

---

[1] I have recieved substantial help with the deep TEX work from Ondřej “Koala” Vácha. Thank you.

Moreover, and it was the bug in the previous version, we have to distinguish between single quote and double quote, where each of them should be printed as different character. Now, it is getting to be really messy `:-)`.

```
\iffalse
   \catcode96=13
   \def``{{\usefont{OT1}{cmss}{m}{n}\symbol{92}}}
   \catcode39=13
   \def'{\protect\MS@quote}
   \def\MS@quote{\futurelet \nextchar \MS@questquote}
   \def\MS@questquote{\ifx `\nextchar \let\MS@next=\MS@dblq
      \else \let\MS@next=\MS@sglq
      \fi \MS@next}
   \def\MS@dblq{{\usefont{OT1}{cmss}{m}{n}\symbol{125}}}
   \def\MS@sglq{{\usefont{OT1}{cmss}{m}{n}\symbol{39}}}
\fi
```

`\sfdefault` `\ttdefault`

Of course, in typewritter there are no different fonts for sans-serif characters and there is no distinction between normal and `tt` characters, thus both of these are set to be same as roman characters.

```
\renewcommand{\sfdefault}{\rmdefault}
\renewcommand{\ttdefault}{\rmdefault}
```

`typearea`

Of course, typewritter should follow classical “one inch on all sides” margins (later, variant supporting European equivalent on A4 paper may be added). The best is to use special package `fullpage`. However, that does not work well with `koma-script` family of packages, there we have to distinguish between the two and use macro `\typearea` native of `koma-script`.

```
\@ifundefined{typearea}
   {\RequirePackage{fullpage}}
   {\typearea[0pt]{13}}
```

`\textbf` `\bfseries` `\emph` `\em`

Italic characters are not enough distinctive in `cmtt` font, so we shall redefine macro `\emph` to be same as in the typewritter age—underlining characters.

```
\renewcommand{\textbf}[1]{\textsc{#1}}
\def\bfseries{\scshape}

\AtBeginDocument{%
   \setul{0.3ex}{0.15ex}
   \renewcommand{\emph}[1]{\ul{#1}}
   \def\em{\egroup \expandafter \ul \expandafter{\iffalse}\fi}
}
```

`\spaceskip`

There are some additional characteristics which should be set-up emulating behavior of typewritter: footnotes should be set in the normal-size font, document should be double-spaced, ragged right. Space between characters have to be set via `\spaceskip`—it is zero on default for `tt` fonts.

```
\let\footnotesize\@empty
\doublespacing
```

```
45 \AtBeginDocument{%
46   \raggedright
47   \parindent 1em
48   \spaceskip .333333 em plus .333333 em minus .111111 em }
```

`\thefootnote`
`\@makefnmark`
`\@makefntext`

If the idea of the whole package is to emulate typewriter style, then we have to do something about footnotes. There is obviously nothing like superscripted footnote mark on typewriters. Moreover, I have also changed indentation of the body of footnote.

```
49 \def\thefootnote{\@arabic\c@footnote/}
50 \def\@makefnmark%
51   {\hbox{\normalfont\@thefnmark}}
52 \renewcommand\@makefntext[1]{%
53    \leftskip 1.8em \noindent
54    \llap{\normalfont\@thefnmark\ }#1}%
```

`\@maketitle`

The standard titlehead of the document is really ugly when doublespaced. Therefore we should redefine `\@maketitle` macro. However, the trick below (creation of new macro which envelopes the original macro) is better, because it is compatible both with standard `article`-like classes and `koma-script` family. Redefinition of `\huge` is a kind of ugly hack, but it should be enough robust and simple to be OK.

```
55 \let\MS@maketitle=\@maketitle
56   \if@titlepage
57      \def\@maketitle{%
58         \hyphenpenalty=5000
59         \let\huge\LARGE
60         \MS@maketitle }
61   \else
62      \def\@maketitle{%
63         \singlespacing
64         \hyphenpenalty=5000
65         \let\huge\LARGE
66         \MS@maketitle }
67   \fi
```

That's all folks! `:-)`

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Change History

1.0
- General: Initial version 1

1.2
- General: `\fileversion` and `\filedate` which does not seem to work at all being replaced by `\RCS` command from `rcs` package. 1

1.4
- General: With help of Stepan Kasal (stepan at matsrv dot mat dot cas dot cz) I have managed to get `\fileversion` working, so I have get rid off rcs.sty again. . . 1
- `\sectfont`: Courier is better, so that this hack is better to be used only for `cmtt` font. 2

1.5
- `\@makefntext`: Whole redefinition of footnotes added. 4
- `\@maketitle`: When `\titlepage` option is on, we need not singlespacing. 4
- `\rmdefault`: With CM-Super we need not `\ae` package anymore. 2

1.6
- General: Some typos corrected. . . 1

1.7
- General: New upload to CTAN and fix TEX logo 1