

altfont: Using alternative fonts*

Sebastian Marius Kirsch
sebastian_kirsch@kl.maus.de

1996/07/02

Abstract

With this package, you can use many alternative fonts with one single package. `psNFSS` and `mfNFSS` use one file for every font, while I prefer having everything in one package. Besides, if you load two of the font changing packages, the latter one overwrites the changes of the first one. Having all in one package avoids this.

1 Introduction

This package is intended as a replacement for some parts of both `psNFSS` and `mfNFSS`. It is a replacement for the parts that actually change the default fonts. It was originally written as a replacement for `psfonts` of `psNFSS`, but I realized that it was not very reasonable to restrict it to PostScript fonts. So it is now not focused on any particular PostScript font, but can be used for any font, METAFONT or PostScript. I'll discuss this in detail in section 4.

2 Installation

The actual package is produced by running `TEX` on `altfont.ins`. This produces `altfont.sty`, the style file, and `altfont.cfg`, the configuration file. Both files must be moved where `TEX` can find them.

2.1 The configuration file

`\AvailableRMFont` All available fonts must be declared in the configuration file. This is done using the macros `\AvailableRMFont`, `\AvailableSFFont`, `\AvailableTTFont`, and `\AvailableFont`. They have the following syntax:

```
\AvailableRMFont[<additional code>]{<long name>}{<NFSS family name>}  
\AvailableSFFont[<additional code>]{<long name>}{<NFSS family name>}  
\AvailableTTFont[<additional code>]{<long name>}{<NFSS family name>}  
\AvailableFont[<additional code>]{<symbolic name>}{<roman font>}%  
  {<sans serif font>}{<typewriter font>}
```

*This file has version number 1.1, last revised 1996/07/02.

The first argument contains a symbolic name for the font, while the second argument contains the internal name for the family. The symbolic name is to be used as an option to the package. (It is easier to remember Times as `times` than as `ptm`.)

The optional argument contains code that is executed after the font is loaded. It can be used to e. g. change the font encoding or to redefine some other defaults such as `\bfdefault`.

The macro `\AvailableFont` does the same, but for three font families at once. This can be used for font families that have the variants sans serif and typewriter, e. g. the Computer Modern fonts, as well as for managing a combination of three fonts under one short name. The arguments can also be empty.

Typical commands look like this:

```
\AvailableRMFont{times}{ptm}
\AvailableSFFont{helv}{phv}
\AvailableTTFont{Courier}{pcr}
\AvailableFont[\altfontenc{T1}]{dco}{cmor}{cmoss}{cmott}
```

`\DefaultRMFont` You can also tell `altfont` about default PostScript fonts. If the user calls the option `onlyps`, the default fonts are substituted for the fonts he has not defined.
`\DefaultSFFont` This is useful when you do not want to supply every font to be used in the document, but want your document to contain only PostScript fonts. The syntax of these commands looks like this:
`\DefaultTTFont`

```
\DefaultRMFont{<NFSS family name>}
\DefaultSFFont{<NFSS family name>}
\DefaultTTFont{<NFSS family name>}
```

`\altfontenc` If you want to change the font encoding in the *<additional code>* section of any of the `\Available...Font` macros, you have to use `\altfontenc` instead of a plain `\RequirePackage[...]{fontenc}`. The reason for this is discussed later. This is the syntax of `\altfontenc`:

```
\altfontenc{<encoding>}
```

Generally, you can neither use commands that have optional arguments nor load classes or packages in the *<additional code>* section.

3 Syntax

You load the package with `\usepackage[...]{altfont}`, as usual, with the fonts you want to use in square brackets. Typical commands look like this:

```
\usepackage[palatino,gill,courier]{altfont}
\usepackage[newcent,onlyps]{altfont}
\usepackage[dco]{altfont}
\usepackage[pandora,courier]{altfont}
```

4 psNFSS vs. mfNFSS

There are two other popular packages for using alternative fonts with \LaTeX : `psNFSS` by Sebastian Rahtz, which is responsible for PostScript fonts, and `mfNFSS` by Frank Mittelbach and Rainer Schöpf, which focuses on METAFONT fonts.

In my opinion, it was not very reasonable to divide the font changing between two packages. The fonts look to \LaTeX all the same, so why make a difference in the packages?

For fonts which are to be used as `\{rm,sf,tt\}default` fonts, both packages have little style files which change these defaults. But if you load two of those packages, the latter overwrites the changes of the first one. These packages all do the same, so why not include them into one style file?

Well, I did this, and the result is easier to use and safer than the packages from `psNFSS` and `mfNFSS`. Because of these improvements, I would like my package to get included into these packages. (I know, I'm unselfish. ;-)

Since `psNFSS` and `mfNFSS` focus on particular fonts, while my package can be used with any font, it is not very likely to be included in either package, though both could make use of it.

5 Legal rubbish

`altfont`: A $\LaTeX 2_{\epsilon}$ package for using alternative fonts

Copyright © 1996 Sebastian Marius Kirsch (sebastian_kirsch@kl.maus.de)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

6 The docstrip modules

This file contains three modules to direct `docstrip` in generating the external files:

<code>driver</code>	A short driver for producing the documentation
<code>package</code>	The package itself
<code>config</code>	The local configuration file

7 Producing the documentation

This short driver can be extracted by `docstrip` to produce the documentation.

```
1 <{*driver}
2 \documentclass{ltxdoc}
3
```

```

4 \usepackage{mflogo}
5 \providecommand{\MF}{\textsf{METAFONT}} % for those without mflogo.sty
6
7 \newcommand*\NFSS{\mbox{\textsf{NFSS}}}
8 \newcommand*\psNFSS{\mbox{\textsf{psNFSS}}}
9 \newcommand*\mfNFSS{\mbox{\textsf{mfNFSS}}}
10 \newcommand*\altfont{\textsf{alt\font}}
11 \newcommand*\PS{Post\font-Script}
12
13 \begin{document}
14
15 \DocInput{altfont.dtx}
16
17 \end{document}
18 </driver>

```

8 The Code

8.1 Introduction

First we have to introduce ourselves.

```

19 (*package)
20 \NeedsTeXFormat{LaTeX2e}
21 \ProvidesPackage{altfont}%
22 [\filedate\space v\fileversion\space loading alternative fonts]

```

8.2 Switches

Next, we define some switches. They are used to determine whether some fonts have already been loaded, so we can warn the user when he tries to load two colliding fonts.

```

23 \newif\ifrm@set\rm@setfalse
24 \newif\ifsf@set\sf@setfalse
25 \newif\iftt@set\tt@setfalse

```

8.3 Default PostScript fonts

<pre> \DefaultRFont \DefaultSFont \DefaultTFont </pre>	<p>These are some commands for changing the default PostScript fonts.</p> <pre> 26 \newcommand{\DefaultRFont}[1]{\renewcommand{\def@rmfont}{#1}} 27 \newcommand{\DefaultSFont}[1]{\renewcommand{\def@sfont}{#1}} 28 \newcommand{\DefaultTFont}[1]{\renewcommand{\def@ttfont}{#1}} 29 \newcommand{\def@rmfont}{} 30 \newcommand{\def@sfont}{} 31 \newcommand{\def@ttfont}{} </pre>
--	---

8.4 Declaring the options

<pre> \AvailableRFont \AvailableSFont \AvailableTFont </pre>	<p>These commands are needed later in the configuration file. There is one command for each font family, ie. roman, sans serif or typewriter.</p> <p>They declare their first argument as an option.</p>
--	--

If the option is called, and the font family has already been defined, a `\PackageError` is reported. If not, the second argument is defined as the appropriate family default.

When the family default has successfully been set, the corresponding switch is turned, so it is not redefined anywhere else in this file.

The optional argument contains some code that is to be executed when the defaults have been set. This is a very useful hook, e. g. when a certain font family is only available in OT1 or Cork encoding, we can execute a `\altfontenc{OT1}` after the font is loaded. This command is a little hack to get around some limitations of `\newcommand` and `\DeclareOption` and is described in section 8.6. A plain `\usepackage` will not work!

```

32 \newcommand{\AvailableRMFont}[3][ ]{%
33   \DeclareOption{#2}{%
34     \ifrm@set\PackageError{altfont}{%
35       \protect\rmfamily\space already defined as \rmdefault
36     }{%
37       You tried to load two roman families at the same time,\MessageBreak
38       e.g. times and palatino}
39     \else\renewcommand{\rmdefault}{#3}\rm@settrue
40     #1
41     \fi}
42   }
43 \newcommand{\AvailableSFFont}[3][ ]{%
44   \DeclareOption{#2}{%
45     \ifsf@set\PackageError{altfont}{%
46       \protect\sffamily\space already defined as \sfdefault
47     }{%
48       You tried to load two sans serif families at the same time,\MessageBreak
49       e.g. gill and helvetica}
50     \else\renewcommand{\sfdefault}{#3}\sf@settrue
51     #1
52     \fi}
53   }
54 \newcommand{\AvailableTTFont}[3][ ]{%
55   \DeclareOption{#2}{%
56     \iftt@set\PackageError{altfont}{%
57       \protect\ttfamily\space already defined as \ttdefault
58     }{%
59       You tried to load two typewriter families at the same time,\MessageBreak
60       e.g. courier and typewriter}
61     \else\renewcommand{\ttdefault}{#3}\tt@settrue
62     #1
63     \fi}
64   }

```

8.5 One command fits all

`\AvailableFont` With the macro `\AvailableFont`, all three defaults are set with one option. This is useful for managing combinations of three fonts under one name, as well as for fonts that have the variants sans serif and typewriter. (The Computer Modern fonts are one of those, and Lucida, I think.)

`\AvailableFont` may also have empty arguments.

```

65 \newcommand*{\AvailableFont}[5] []{%
66   \DeclareOption{#2}{%
67     \def\@temprm{#3}\def\@tempssf{#4}\def\@temptt{#5}
68     \ifx\empty\@temprm\else
69       \ifrm@set\PackageError{altfont}{%
70         \protect\rmfamily\space already defined as \rmdefault
71       }{%
72         You tried to load two roman families at the same time,
73         \MessageBreak e.g. times and palatino}
74       \else\renewcommand{\rmdefault}{#3}\rm@settrue
75       \fi
76     \fi
77     \ifx\empty\@tempssf\else
78       \ifssf@set\PackageError{altfont}{%
79         \protect\sffamily\space already defined as \sfdefault
80       }{%
81         You tried to load two sans serif families at the same time,
82         \MessageBreak e.g. gill and helvetica}
83       \else\renewcommand{\sfdefault}{#4}\sf@settrue
84       \fi
85     \fi
86     \ifx\empty\@temptt\else
87       \iftt@set\PackageError{altfont}{%
88         \protect\ttfamily\space already defined as \ttdefault
89       }{%
90         You tried to load two typewriter families at the same time,
91         \MessageBreak e.g. courier and typewriter}
92       \else\renewcommand{\ttdefault}{#5}\tt@settrue
93       \fi
94     \fi
95     #1
96   }
97 }

```

8.6 Loading font encodings in \Available...Font

`\altfontenc` Working on this package, I discovered two limitations of `\newcommand` and `\DeclareOption`.

- You cannot use a command with an optional argument in the argument of another command.
- You cannot load a package with `\RequirePackage` or `\usepackage` in the `<code>` section of a `\DeclareOption` command.

I wanted to be able to load different font encodings in the optional argument of the `\Available...Font` command, so I had to work around these limitations. The result is this command:

```

98 \newcommand*{\altfontenc}[1]{\PassOptionsToPackage{#1}{fontenc}%
99   \AtEndOfPackage{\RequirePackage{fontenc}}}

```

By passing the option to `fontenc` with `\PassOptionsToPackage`, I avoid having an optional argument to `\RequirePackage`. By delaying the `\RequirePackage`, I avoid loading a package in an option. Ugly but useful.

8.7 Loading the configuration file

Next, the configuration file is loaded. If it is not found, an error is issued, because the package is quite useless without it.

```
100 \InputIfFileExists{altfont.cfg}{}{%
101   \PackageError{altfont}{%
102     No local configuration file found
103   }{%
104     The altfont package was loaded without a local\MessageBreak
105     configuration file, so it doesn't know which fonts\MessageBreak
106     are available.}
107   }
108 \end{package}
```

8.8 The configuration file

The configuration file looks like this:

8.8.1 Introduction

As usual...

```
109 (*config)
110 \NeedsTeXFormat{LaTeX2e}
111 \ProvidesFile{altfont.cfg}%
112 [\filedate\space v\fileversion\space Local configuration for altfont.sty]
```

8.8.2 Declaring default PostScript fonts

Here, the default PostScript fonts are defined. They are needed later, if the author does not define all fonts he wants to use, but wants his document to contain only PostScript fonts. Strictly spoken, this does not belong to a package for PostScript and METAFONT fonts, but I think it is nonetheless very useful.

```
113 \DefaultRMFont{ptm}
114 \DefaultSFFont{phv}
115 \DefaultTTFont{pcr}
```

8.8.3 Available fonts

Now the available fonts are defined. They are divided into three categories—roman fonts, sans serif fonts and typewriter fonts, as it is done in $\text{\LaTeX} 2_{\epsilon}$ itself.

They are defined using the `\Available...Font` macros. These macros define a symbolic name to be used as an option to the package, and take the name of the NFSS font family as a second argument.

```
116 \AvailableRMFont{times}{ptm}
117 \AvailableRMFont{palatino}{ppl}
118 \AvailableRMFont{newcent}{pnc}
119
120 \AvailableSFFont{helv}{phv}
121 \AvailableSFFont{gill}{pgs}
122
123 \AvailableTTFont{courier}{pcr}
124
```

```

125 \AvailableFont[\altfontenc{T1}]{dco}{cmor}{cmoss}{cmott}
126 \AvailableFont[\altfontenc{OT1}]%
127 \DeclareFontSubstitution{OT1}{pnr}{m}{n}%
128 \renewcommand{\bfdefault}{b}%
129 \renewcommand{\itdefault}{sl}]{pandora}{pnr}{pnss}{pnnt}
130 </config>

```

In case you wonder, the dco fonts are another package I wrote. They consist of the regular dc fonts with oldstyle digits. I didn't release it yet, contact me if you are interested.

And that's the configuration. Of course one could do much more with this configuration file, declaring new options and all. It's probably the biggest hook a package could have. :-)

8.9 onlyps: Using only PostScript fonts

onlyps This option redefines all other family defaults that have not yet been `\renewcommanded` to PostScript fonts. It uses the default fonts as defined in the configuration file. If no default fonts have been defined, an error is reported.

```

131 (*package)
132 \DeclareOption{onlyps}{
133   \ifrm@set\else
134     \ifx\empty\def@rmfont
135       \PackageWarning{altfont}{No default roman font defined!}
136     \else\renewcommand{\rmdefault}{\def@rmfont}\fi
137   \fi
138   \ifsf@set\else
139     \ifx\empty\def@sffont
140       \PackageWarning{altfont}{No default sans serif font defined!}
141     \else\renewcommand{\sfdefault}{\def@sffont}\fi
142   \fi
143   \iftt@set\else
144     \ifx\empty\def@ttfont
145       \PackageWarning{altfont}{No default typewriter font defined!}
146     \else\renewcommand{\ttdefault}{\def@ttfont}\fi
147   \fi
148 }

```

8.10 The end

At last, the options are processed.

```

149 \ProcessOptions
150 </package>

```