The 'multicap' LaTeX 2_ε package*

John Vassilogiorgakis
giannis@iris.math.aegean.gr

12th May 2002

Abstract

This is a package for formating captions of column figures and column tabular material which cannot be floats (i.e. outside a `figure` or `table` environment in standard LaTeX) in the `multicols` environment provided by the multicol package. It also provides an easy way to customize your captions, either in single column or inside `multicols`.

Contents

1 Introduction

As it is known, the multicol package supports partially floats inside `multicols` environment. To be exact, only the star versions of the floating environments are supported, excluding the existence of column floats—i.e. only page wide floats can be used [3]. However, one can put a column image or tabular material with one or another way—with an `\includegraphics` command, or a `tabularx` environment inside a `center` environment—getting satisfactory results like the following:

	$\mathbf{x_1}$	$\mathbf{x_2}$
1.	$(x-1, y-1)$	$(x+1, y+1)$
2.	$(x, y-1)$	$(x, y+1)$
3.	$(x+1, y-1)$	$(x-1, y+1)$
4.	$(x-1, y)$	$(x+1, y)$

The same thing can happen, of course, with 'column figures'. The column material is placed outside of a floating environment by simply specifying:

... text before column material ...

```
\begin{center}
\begin{tabularx}{\linewidth}{...}
 ...
\end{tabularx}
\end{center}
```

... text after column material ...

The only thing left, is to add captions to this column material, as we would have done if this material were inside a floating environment. Moreover, we want the numbering of these captions to be handled

*Version number v1.0. Last revised 2002/05/04.

correctly, even if a star-form of a floating environment appears and even if we are switching between `multicols` and single column layout. Finally, we also want the text specified by the caption to appear in the list of figures or tables and the references to this material to work correctly.

2 Usage

The package loads with

```
\usepackage{multicap}
```

and it requires the ifthen package. The multicap package defines the `\mfcaption` and `\mtcaption` commands, which replace the `\caption` command outside of a floating environment; for column figures and column tabular material respectively, that are necessary outside of a floating environment inside `multicols`.

The syntax of the commands is the same as the one of `\caption`'s:

```
\mfcaption[short text]{long text}
```

for formating captions of 'column figures' and

```
\mtcaption[short text]{long text}
```

for formating captions of column tabular material ('column tables'). As in the case of the `\caption` command, the optional argument *short text* goes into the list of figures or tables. If this is not specified, then the mandatory argument *long text* does the job.

Thus, the previous 'orphan' tabular[1] can now have a caption like this:

	$\mathbf{x_1}$	$\mathbf{x_2}$
1.	$(x-1, y-1)$	$(x+1, y+1)$
2.	$(x, y-1)$	$(x, y+1)$
3.	$(x+1, y-1)$	$(x-1, y+1)$
4.	$(x-1, y)$	$(x+1, y)$

Table 1: Caption inside `multicols`.

The command line arguments were:

```
\begin{center}
\begin{tabularx}{\linewidth}{...}
 ...
\end{tabularx}
\mtcaption[This captions produced with the
          \mtcaption command.]{Caption inside
          \texttt{multicols}.}
\end{center}
```

[1]Sorry for using the same example again and again.

It is clear now, that you just put the column material—either this is a tabular material, a figure, a picture loaded with an `\epsfig`, `\includegraphics` or anything relative—followed by an `\mfcaption`, for column 'figures', or an `\mtcaption` for 'column tables', command inside a `center` environment and thats all. You have a caption for this column material, as if it was inside a floating environment. Moreover, placing the whole stuff inside a `minipage` and using the `\columnbreak` command, you can achieve good looking results by manually placing the column float at the top or the bottom of the page.

2.1 Options and customization

A tactic of many well known typographers and something that one encounters very often in well printed books, is that the comments explaining what is shown in a floating element, are typesetting with a sans serif font and with a completely different lead (baseline skip) and font size than the main document's. For example, you may have a book typeseted in 10/12, and the captions of the floats to be in 9/11.5 or something else, depending from the kind of the font. Especially the use of unmodulated sans serif fonts with a smaller font size than the main document's and the appropriate lead, is something that harmonizes well enough with the narrow linewidth of the two column output.

2.1.1 The `normal` option

The captions produced by the `\mfcaption`, `\mtcaption` and `\caption` commands with multicap, are typesetting the `\figurename` (or `\tablename`) and the mandatory argument *long text* with the sans serif font defined by the `\sfdefault` command. The number that is associated with the column float is typeseted with SMALL CAPS, if there exists this shape for the sans. If not, a

```
\renewcommand{\thefigure}{%
 \textmormal{\textsc{\arabic{figure}}}}
```

in the preamble solves the problem[2], typesetting the counter with small caps but with the normal font this time. This is to deal with the high level typographic demands, that wants the comments explaining what is shown in a float to be typeseted with sans and the number associated with this floating element to be old style.

If you don't want this perspective, multicap offers the unique package option `normal` that typesets all captions in the usual way. This can be chosen by specifying

```
\usepackage[normal]{multicap}
```

in the preamble.

2.1.2 Changing caption's \fontsize parameters

Package `multicap` offers you also the possibility to easily change the `\fontsize` parameters of the captions. This can be done by simply changing the values of the

`mcapsize` and `mcapskip`

counter variables with the `\setcounter` declaration. What really happens, is that the values of the above counters go into the first and second argument respectively of the `\fontsize` command, that changes locally inside `\mfcaption`, `\mtcaption` and `\caption`. For example, if you want your captions to be typeseted in 9/11, then you have to do in the preamble the following declarations:

```
\setcounter{mcapsize}{9}
\setcounter{mcapskip}{11}.
```

We should emphasize at this point, that multicap's options, along with the `mcapsize` and `mcapskip` parameters, take effect *and* in the `\caption` command which the package redefines. Thus, the usage of multicap is somehow wider, in the sense that these tuning options were not given especially for a multicolumn layout. The truth is that in multiple columns such changes fit better, but a novice user can also use them to format his captions of his single column layout. This is not true for the parameters discussing in the next sections, which act affects only the two new commands `\mfcaption` and `\mtcaption` and their usage is restricted only for multicolumn output.

2.1.3 The \abvmcapskip length

The vertical space added between the column float and the caption produced either by the `\mfcaption` or `\mtcaption` command, is controlled by the

`\abvmcapskip`

length parameter, which is equal to the value of 10pt. It is not suggested to change this value (with the `\setlength` declaration), unless if you want to typeset captions with an extremely different font size and `\baselineskip` than the main document's.

2.1.4 The \blwmcapskip length

From the other side now, if you decide not to use some of the `\mfcaption` and `\mtcaption` commands that appear in your code, inside a `center` environment[3] (i.e. you may want to put them inside the range of a `\centering` command or anything of the above, thus giving a more compact look on your multicolumn layout), there is the possibility of changing the vertical space separating the caption from the following text by changing the

`\blwmcapskip`

length parameter (again with `\setlength`). The default value of `\blwmcapskip` is equal to the value of `\parsep`. However, such a change will affect only the captions produced by the `\mfcaption` and `\mtcaption` commands that are outside of a `center` environment.

This tuning option is made possible, because placing one of the `\mfcaption` or `\mtcaption` declarations outside of a `center` environment, the vertical space added between the *text before column material* and the column float, will be smaller than the vertical space that follows after the caption and before the *text after the column material*. This way, with the `\blwmcapskip`, you can balance the vertical space before and after the column float.

In any case, if you wish to have a more compact look using the `\centering` command for example, you should specify something like this:

... text before column material ...

```
\par{\centering
\begin{tabularx}{\linewidth}{...}
...
\end{tabularx}}
\mtcaption{Another caption inside multicols.}
```

[2]Replace `\textsc` with `\oldstylenums` if you are typesetting with the Computer Modern fonts.
[3]I suggest not to do so.

... text after column material ...

leaving the `\mfcaption` or `\mtcaption` declarations outside the range of `\centering`[4].

	$\mathbf{x_1}$	$\mathbf{x_2}$
1.	$(x-1, y-1)$	$(x+1, y+1)$
2.	$(x, y-1)$	$(x, y+1)$
3.	$(x+1, y-1)$	$(x-1, y+1)$
4.	$(x-1, y)$	$(x+1, y)$

Table 2: Another caption inside `multicols`.

2.2 Numbering

When formating captions in `multicols` using the multicap package and a star-form of a floating environment appears, the `\caption` command keeps numbering the figure or table that spans all columns now correctly, continuing from where a `\mfcaption` or `\mtcaption` command had stopped, depending from the kind of float. The same thing happens when the opposite occurs—when switching from the page-wide float to the column float—and when switching from `multicols` to single column and back. This happens because the `\mfcaption` uses the `figure` counter to count the column figures and the `\mtcaption` command the `table` counter to number the column tables. So the interchange between `\mfcaption`, `\mtcaption` and `\caption` and between one and multicolumn mode, does not affect the numbering of the figures or tables that are handled correctly from the package.

	$\mathbf{x_1}$	$\mathbf{x_2}$
1.	$(x-1, y-1)$	$(x+1, y+1)$
2.	$(x, y-1)$	$(x, y+1)$
3.	$(x+1, y-1)$	$(x-1, y+1)$
4.	$(x-1, y)$	$(x+1, y)$

Table 3: Table produced by the `table` environment.

3 Implementation

```
\NeedsTeXFormat{LaTeX2e}%
\ProvidesPackage{multicap}[2002/05/04
 v1.0 formatting captions inside multicols]%
```

The package requires the ifthen package to test some conditions.

```
\RequirePackage{ifthen}%
```

We now define the `\@mcaptype` command which will switch between the normal font and the Sans. Next, we define the default (`Sans`) and the the `normal` option, together with a warning for any unknown option to the package.

```
\newcommand*{\@mcaptype}{\@empty}%
\DeclareOption{Sans}%
  {\renewcommand*{\@mcaptype}{\textsf}}%
\DeclareOption{normal}%
  {\renewcommand*{\@mcaptype}{\relax}}%
\DeclareOption*{%
 \PackageWarning{multicap}{Unknown option
 `\CurrentOption'.\MessageBreak
 Going into default mode}%
}%
\ExecuteOptions{Sans}%
\ProcessOptions*%
```

The counter variables '`mcapsize`' and '`mcapskip`' which go to first and second argument of the `\fontsize` command respectively are defined. The `\abvmcapskip` and `\blwmcapskip` lengths are defined too and their default values are being given.

```
\newcounter{mcapsize}%
\newcounter{mcapskip}%
\newlength{\abvmcapskip}%
\newlength{\blwmcapskip}%
\setlength{\abvmcapskip}{10\p@}%
\setlength{\blwmcapskip}{\parsep}%
```

We define one box for each command `\mfcaption`, `\mtcaption` and `\caption` to make some length tests later.

```
\newsavebox{\mf@tempbox}%
\newsavebox{\mt@tempbox}%
\newsavebox{\@temp@box}%
```

We redefine the `center` environment so as to set the boolean `@cent@er` true when the environment starts and false when it ends.

```
\newboolean{@cent@er}%
```

[4] We would have done the same thing, if we were using the `\caption` command.

```
\renewenvironment*{center}{%
 \setboolean{@cent@er}{true}\trivlist%
 \centering\item\relax}%
 {\setboolean{@cent@er}{false}\endtrivlist}%
```

The command `\@mcap@fs` checks weather the values of the 'mcapsize' and 'mcapskip' have change, sends them to `\fontsize` and stores them in the `\@mcap@size` command.

```
\DeclareRobustCommand*{\@mcap@size}{\relax}%
\newcommand*{\@mcap@fs}{%
 \ifthenelse{\equal{\value{mcapsize}}{0}%
  \and\equal{\value{mcapskip}}{0}}{}%
  {\renewcommand*{\@mcap@size}{%
    \fontsize{\value{mcapsize}}{%
     \value{mcapskip}%
    }%
    \selectfont%
   }%
  }%
}%
```

We are now ready to define the `\mfcaption` command. After an initial skip of `\abvmcapskip` (10pt) we increment the `figure` counter with `\refstepcounter` so that the references to the column figures will work correctly. We check with `\@mcap@fs`, if the values of the 'mcapsize' and 'mcapskip' counters have been changed and next we add the text specified in the `\mfcaption` command to the list of figures.

```
\newcommand*{\mfcaption}[2][\@empty]{%
 \\[\abvmcapskip]%
 \@mcap@fs%
 \refstepcounter{figure}%
 \ifthenelse%
  {\equal{\protect#1}{\protect\@empty}}%
  {\addcontentsline{lof}{figure}%
   {\protect\numberline{\thefigure}#2}}%
  {\addcontentsline{lof}{figure}%
   {\protect\numberline{\thefigure}#1}}%
```

After the mandatory argument *long text* has been stored in `\mf@tempbox`, we compare the width of the *long text* with that of `\linewidth`. If the comments of the figure (*long text*) fit on a line, then they will be centered, otherwise they will be typeset as a paragraph with width `\linewidth`.

```
 \sbox{\mf@tempbox}{%
  \@mcap@size\@mcaptype{%
   \figurename\ \textsc{\thefigure{}:} #2%
  }%
 }%
 \ifthenelse{\lengthtest{%
  \wd\mf@tempbox > \linewidth}}%
  {\begin{minipage}[t]{\linewidth}%
   \@mcap@size\@mcaptype{%
   \figurename\ \textsc{\thefigure{}:} #2}%
   \end{minipage}\par%
  }%
 {{\centering\usebox{\mf@tempbox}\\}}%
```

An skip of `\blwmcapskip` follows, if we are not inside a `center` environment.

```
 \ifthenelse{\boolean{@cent@er}}{}%
  {\vspace{\blwmcapskip}}%
}%
```

The `\mtcaption` command is defined by exact the same way.

```
\newcommand*{\mtcaption}[2][\@empty]{%
 \\[\abvmcapskip]%
 \@mcap@fs%
 \refstepcounter{table}%
 \ifthenelse%
  {\equal{\protect#1}{\protect\@empty}}%
  {\addcontentsline{lot}{table}%
   {\protect\numberline{\thetable}#2}}%
  {\addcontentsline{lot}{table}%
   {\protect\numberline{\thetable}#1}}%
 \sbox{\mt@tempbox}{%
  \@mcap@size\@mcaptype{%
   \tablename\ \textsc{\thetable{}:} #2%
  }%
 }%
 \ifthenelse{\lengthtest{%
  \wd\mt@tempbox > \linewidth}}%
  {\begin{minipage}[t]{\linewidth}%
   \@mcap@size\@mcaptype{%
   \tablename\ \textsc{\thetable{}:} #2}%
   \end{minipage}\par%
  }%
 {{\centering\usebox{\mt@tempbox}\\}}%
 \ifthenelse{\boolean{@cent@er}}{}%
  {\vspace{\blwmcapskip}}%
}%
```

Finally, we redefine the kernel `\@makecaption` command, so as `\caption` can have the same 'properties' with the `\mfcaption` and `\mtcaption` commands.

```
\renewcommand{\@makecaption}[2]{%
 \vspace{10\p@}%
 \@mcap@fs%
 \sbox{\@temp@box}%
  {\@mcap@size\@mcaptype{#1{}\textsc{:} #2}}%
 \ifthenelse{\lengthtest{%
  \wd\@temp@box > \linewidth}}%
  {\@mcap@size\@mcaptype{#1{}\textsc{:} #2}%
   \par%
  }%
  {\centering\@mcap@size\@mcaptype{%
   #1{}\textsc{:} #2%
   }%
  }%
}%
\endinput
```

List of Tables

References

[1] M. Goossens, F. Mittelbach and A. Samarin, "*The LaTeX Companion*", Addison Wesley, 1994.

[2] The LaTeX3 Project, "*LaTeX 2ε for class and package writers*", 1995–1998.

[3] Frank Mittelbach, "*An environment for multicolumn output*", CTAN, file *multicol.dtx*, March, 2001.

[4] R. Bringhurst, "*The Elements of Typographic Style*", Hartley & Marks, 1996.