Package `graphfig`*

F. Bosisio
E-mail: `fbosisio@bigfoot.com`

1997/12/15

Abstract

Documentation for the package `graphfig`.

1 Introduction

This package provides some commands to make the use of graphic files in LaTeX simpler.
It declares the "`Figure`" environment (capitalized!) and the two commands "`\graphfile`" and "`\graphfile*`". Combining this commands, it it possible to include graphic files in a LaTeX document very simply.

2 Required packages

This package uses the "`\includegraphics*`" command defined in the standard `graphics` package. Moreover, it uses the package `subfigure` when the `subfigure` option is specified, and the `float` package if the `AllowH` option is used.

3 The options

At now, two options are available: "`subfigure`" and "`AllowH`".
The "`subfigure`" option allow the use of sub-figures inside a `Figure` environment, in order to place multiple pictures in a single LaTeX figure (cfr. the `subfigure` standard package).

*This is version 2.2, last revised 1997/12/15; documentation date 2005/04/09

The "AllowH" option allow the use of the "H" float specifier for the Figure environment, in order to place the figure exaclty where the command is placed (cfr. the float standard package).

4 The Figure environment and its relatives

The "Figure" environment (capitalized!) is somewhat different from the standard LaTeX "figure" environment: besides an optional argument used to specify the placement parameters (which now defaults to "[htbp]"), it has a mandatory argument specifying the "caption" and another optional argument, used as a "label" for cross-referencies:

```
\begin{Figure}[<htbpH>]{<caption>}[<label>]
            ...
\end{Figure}
```

The use of the "H" specifier (i.e. "I want my float here!") is possible only if the "AllowH" option has been specified.

4.1 The graphfile command

Inside the "Figure" environment, are available the commands:

```
\graphfile[<width>]{<file>}[<sub-caption>]
\graphfile*[<heigth>]{<file>}[<sub-caption>]
```

which are a simplified version of the "\includegraphics*" command (which is automatically included by this package; see the graphics package for reference), since you don't have to worry about scaling: the mandatory argument is the name of the graphic file to include, whereas the first optional argument specifies the desired width (in the non *-form) or heigth (in the *-form) of the figure as a fraction of "\linewidth" (e.g. "50" means ".50\linewidth", i.e. half a line!), so no unit of measure (as "cm" or "pt") is required. Moreover, since the "*-form" of "\includegraphics" is used, the regions outside the bounding-box are not drawn. Another advantage of the combined use of the "Figure" environment and the "\graphfile" commands is that the picture is automatically centered horizontally, so no "\centering" or similar declartion is required.

```
\begin{Figure}[<htbpH>]{<caption>}[<label>]
        \graphfile[<width>]{<file>}
\end{Figure}
```

4.2 Sub-figures

If you want to include more than one file in a single figure, you need to specify the "`subfigure`" option, which includes the `subfigure` package and provides the authomatic placement of a sub-caption below each picture.

```
\graphfile[<width_1>]{<file_1>}[<sub-caption_1>]
...
\graphfile[<width_N>]{<file_N>}[<sub-caption_N>]
```

which is a combined version of "`\subfigure`" and "`\includegraphics*`" (see the `subfigure` and `graphics` packages for reference): the last optional argument specify a "caption" for the sub-figure under consideration, while the first two arguments work exactly like as described above in the case of one only picture (indeed, you can use the last optional argument even if the `subfigure` option was not specified, in which case it is simply ignored). Each individual caption is printed preceded by a bracketed letter, which is the sub-figure counter and is printed even if no caption is specified. Using only a series of "`\graphfile[...]{...}[...]`" commands inside a "`Figure`" environment provides an equal spacing beetween the pictures and around them, without the need for any extra command. Finally, if a "`<label>`" was specified as the last optional argument to the "`Figure`" environment, you can reference to each individual sub-figure by the labels "`<label>:a`", "`<label>:b`", and so on, without the need for declaring them.

```
\begin{Figure}[<htbpH>]{<caption>}[<label>]
\graphfile[<width_1>]{<file_1>}[<sub-caption_1>]
\graphfile[<width_2>]{<file_2>}[<sub-caption_2>]
       :
       :
\end{Figure}
```

4.3 The `FigureDefaultPlacement` command

The command "`FigureDefaultPlacement{...}`" can be used to specify the default value for the placement argument of the "`Figure`" environment. This is useful to change the default placement from the standard value "htbp".

5 Implementation

We start defining the options and including the required packages.

```
%%
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{\FileName}[\filedate\space v\fileversion\space\filedescr]
\RequirePackage{graphics}
%%
\newif\if@AllowSubFigure\@AllowSubFigurefalse
\newif\if@AllowHfloat\@AllowHfloatfalse
%%
\DeclareOption{subfigure}{\@AllowSubFiguretrue}
\DeclareOption{AllowH}{\@AllowHfloattrue}
%%
\ProcessOptions
%%
\if@AllowSubFigure
  \RequirePackage{subfigure}[1995/03/06 v2.0]
```

These redefinitions are needed in order to obtain the right form of the references (i.e. like "Fig. 1a" and not "Fig. 1(a)")

```
  \def\thesubfigure{\alph{subfigure}}
```

We use "`\renewcommand`" instead of "`\def`" in order to realize possible changes in the `subfigure` package which may eliminate the inner command "`\@thesubfigure`".

```
  \renewcommand*\@thesubfigure{{\subcaplabelfont(\thesubfigure)}\space}
  \let\SubGR@PH=\subfigure
\else
```

If the "subfigure" option is not present, we let the internal command "`\SubGR@PH`" to simply execute its mandatory argument

```
  \newcommand*\SubGR@PH[2][]{#2}
\fi
\if@AllowHfloat
  \RequirePackage{float}[1995/03/29 v1.2c]
  \restylefloat{figure}
\fi
%%
\newif\if@FirstPicture\@FirstPicturetrue
\let\SubFig@Label=\relax
```

`\@Graph@Figure` This is the core command which includes and scales the graphic file using the commands provided by the `graphic` package.

```
%%
\newcommand*\@Graph@Figure[3]{\resizebox{#1}{#2}{\includegraphics*{#3}}}
```

`\graphfile` The command "`\graphfile''` and its *-version simply calls one of the forthcoming commands.

```
\newcommand*\graphfile{\@ifnextchar*{\graphfile@star}{%
\@ifnextchar[{\GraphFile@width}{\GraphFile@noSize}}%
}
\def\graphfile@star*{\@ifnextchar[{\GraphFile@heigth}{\GraphFile@noSize}}
```

These commands, in turns, calls "`\@GraphFile@Draw`" with suitable arguments (i.e. the lengths are turned into fractions of "`\linewidth`" and an exclamation mark "`!`" is used where the scaling factor has to be the same as the other direction; furthermore the actual size of the graphic image is used [throught the "`\width`" command] if no scaling parameter is given).

```
%%
\def\GraphFile@width[#1]{%
\@ifnextchar[{\@GraphFile@width@heigth{#1}}{\@GraphFile@widthNOheigth{#1}}%
}
\def\@GraphFile@width@heigth#1[#2]#3{%
\@ifnextchar[{\@GraphFile@Draw{.#1\linewidth}{.#2\linewidth}{#3}}{%
\@GraphFile@Draw{.#1\linewidth}{.#2\linewidth}{#3}[]}%
}
\def\@GraphFile@widthNOheigth#1#2{%
\@ifnextchar[{\@GraphFile@Draw{.#1\linewidth}{!}{#2}}{%
\@GraphFile@Draw{.#1\linewidth}{!}{#2}[]}%
}
\def\GraphFile@heigth[#1]#2{%
\@ifnextchar[{\@GraphFile@Draw{!}{.#1\linewidth}{#2}}{%
\@GraphFile@Draw{!}{.#1\linewidth}{#2}[]}%
}
\newcommand*\GraphFile@noSize[1]{%
\@ifnextchar[{\@GraphFile@Draw{\expandafter\width}{!}{#1}}{%
\@GraphFile@Draw{\expandafter\width}{!}{#1}[]}%
}
```

`\@GraphFile@Draw` This command has three mandatory arguments and an optional one, which represent, respectively, the width, heigth, file-name and sub-caption of the current picture.

```
%%
\def\@GraphFile@Draw#1#2#3[#4]{%
\if@FirstPicture%
```

If it is the first call to this command in the current `Figure` environment, we simply store it in "`\TMP@Graph`". We also set "`@FirstPicture`" to false for the next call.

```
  \@FirstPicturefalse%
  \def\TMP@Graph{\SubGR@PH[#4\SubFig@Label]{\@Graph@Figure{#1}{#2}{#3}}}%
\else%
  \ifx\TMP@Graph\undefined%
```

If "`\TMP@Graph`" is not defined, this is the third call (or more). So we invoke the "`\SubGR@PH`" command (which may be "`\subfigure`" or simply its second argument depending on the options) with a "`\@Graph@Figure`" command inside and followed by an "`\hfill`" command for centering purposes.

```
    \SubGR@PH[#4\SubFig@Label]{\@Graph@Figure{#1}{#2}{#3}}\hspace*{\fill}%
  \else%
```

If we arrive at this point it is the second call to "`\graphfile`" inside the current `Figure` environment: before plotting the current figure (as above) we must call "`\TMP@Graph`" to plot the first one (which was only saved). We also undefine "`\TMP@Graph`" for the next time.

```
    \hspace*{-4pt}\TMP@Graph\hspace*{\fill}%
    \let\TMP@Graph\undefined%
    \SubGR@PH[#4\SubFig@Label]{\@Graph@Figure{#1}{#2}{#3}}\hspace*{\fill}%
  \fi%
\fi%
}
```

`DefaultPlacement`

The "`\FigureDefaultPlacement`" macro is defined to set the value of the "`\Default@FigurePlacement`" to its argument. Also "`\Default@FigurePlacement`" is initialized to "`htbp`".

```
%%
\newcommand*\FigureDefaultPlacement[1]{\def\Default@FigurePlacement{#1}}
\def\Default@FigurePlacement{htbp}
```

`Figure`

```
%%
\newenvironment{Figure}[2][\Default@FigurePlacement]{%
```

The code for "`\begin{Figure}`" starts by setting "`@FirstPicture`" to true.

```
\@FirstPicturetrue%
```

Then we start a standard "`figure`" environment followed by an "`\hfill`" command (for centering purposes).

```
\figure[#1]%
\hspace*{\fill}%
```

We change the definition of "\\" so that an "\hfill" command is added at the beginning of next line.

```
\let\@Figure@CR=\\%
\def\\{\par\hspace*{\fill}}%
```

If there is the last optional argument, "\@Figure@quadra" is invoked, otherwise "\MK@Figure@Caption" is directly defined to print the specified caption.

```
\@ifnextchar[{\@Figure@quadra{#2}}{\def\MK@Figure@Caption{\caption{#2}}}%
}{%
```

The code for "\end{Figure}" starts by checking that "\TMP@Graph" is undefined, otherwise this means that there was one only picture: in such a case we have to print it out before continuing.

```
\ifx\TMP@Graph\undefined\else%
  \if@AllowSubFigure%
    \renewcommand*\SubGR@PH[2][]{##2}%
    \TMP@Graph\hspace*{\fill}%
    \let\SubGR@PH=\subfigure%
  \else%
    \TMP@Graph\hspace*{\fill}%
  \fi%
  \global\let\TMP@Graph\undefined%
\fi%
```

The "\MK@Figure@Caption" prints the previously saved caption.

```
\MK@Figure@Caption%
```

All local commands are undefined in order to save TEX memory.

```
\let\Mk@Figure@Caption\undefined%
\let\SubFig@Label=\relax%
\let\\=\@Figure@CR%
\let\@Figure@CR\undefined%
```

The standard "figure" environment is closed.

```
\endfigure%
```

The command "\thesubfigure" is redefined so that it correctly prints the figure number, too.

```
\def\thesubfigure{\thefigure\alph{subfigure}}%
}
```

\@Figure@quadra The "\MK@Figure@Caption" is defined to print the specified caption with an added label. Also "\SubFig@Label" generates a label in the form "<label>:a".

```
%%
\def\@Figure@quadra#1[#2]{%
```

```
\def\MK@Figure@Caption{\caption{#1}\label{#2}}%
\def\SubFig@Label{\expandafter\label{#2:\expandafter\alph{subfigure}}}%
}
```

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