

The `nccpic` package*

Alexander I. Rozhenko
rozhenko@oapmg.sssc.ru

2005/03/06

1 User Interface

The package is considered as an extension of the `graphicx` package. It passes all options to the `graphicx` and customizes graphics extensions list for most popular drivers, namely `dvips` and `dvidpf`.

This allows us omit an extension of a graphics file in the `\includegraphics` command. When a file without extension is searched, this command sequentially tries extensions from the list of extensions until an appropriate file will be found.

Using this feature, you can support multiple output from a \LaTeX document with minimum changes in `.tex` sources. The only required thing is to prepare a number of versions for all graphics files used in the document. For example, the `dvips` program and YAP previewer like `.eps` and `.bmp` files, and the `pdftex` likes `.png` files. To satisfy their needs, you can prepare the `.eps` or `.bmp` version of all pictures for use with `dvips` and the `.png` version for use with `pdftex`. Then, when you translate your source file with the `latex` command, the `.eps` or `.bmp` versions of graphics files are used. But when you translate your source file with the `pdflatex` command, the `.png` versions of graphics files are used.

The next aim of this package is the regulation of placement of graphics files in the file system. It is the bad idea to place graphics together with \LaTeX sources, especially when you prepare a book containing many pictures. It will be much better if graphics will be stored in a subdirectory relative to the base directory of your \LaTeX sources. We propose to store graphics files in the `graphics` subdirectory of the source directory. To support the search in this storage, the graphics path is customized in this package.

`\ipic` The graphics in \LaTeX can be prepared at least in two ways: as an external graphics file included with the `\includegraphics` command or its analogue; as a \LaTeX `picture` environment or its analogues provided with special graphics packages. In the last case, the placing a picture inside a \LaTeX source file is inconvenient. It will be better to put a \LaTeX picture in a separate file and include it using the `\input` command. The next step is to place such pictures together

*This file has version number v1.2, last revised 2005/03/06.

with other graphics material into the **graphics** subdirectory and load them using the `\ipic{⟨filename⟩}` command. This command appends the `⟨filename⟩` with the `.pic` extension and tries to load the `⟨filename⟩.pic` file at first from the base directory of your document and then from the **graphics** subdirectory.

`\draftgraphics`
`\finalgraphics`

The `\draftgraphics` and `\finalgraphics` commands respectively set the draft and final modes for graphics inclusion with the `\includegraphics` command. They are analogues for the **draft** and **final** options of the **graphics** package. But the commands allow more flexible control of graphics mode especially for document classes including this package by default (the **ncc** class from **NCC-L^AT_EX** does this). A graphics file included in the draft mode is shown as a rectangle containing the name of graphics file. This is quite useful on the stage of editing a document.

`\putimage`

The following command includes a graphics file using the `\includegraphics` command. It is provided for compatibility with old **NCC-L^AT_EX**:

$$\text{\texttt{\textbackslash putimage}}(x,y)[x_{\text{real}},y_{\text{real}}](x_{\text{shift}},y_{\text{shift}})\{\text{\textit{⟨filename⟩}}\}$$

Here:

`⟨filename⟩` is a name of graphics file;

`(x,y)` are dimensions (in `\unitlength`) of the prepared image box;

`[xreal,yreal]` are dimensions (in `\unitlength`) of graphics file that are passed to the `\includegraphics` command;

`(xshift,yshift)` are shifts (in `\unitlength`) of the graphics image with respect to the image box.

The `(x,y)` and `⟨filename⟩` parameters are required. Others are optional. If the `[xreal,yreal]` parameter is omitted, $x_{\text{real}} := x$ and $y_{\text{real}} := y$. If the `(xshift,yshift)` parameter is omitted, the shift values are set to zero.

The `\putimage` command vertically aligns the image box in the different way than the `\includegraphics`. The first one lowers the box in such a way to align its top edge with the top edge of letter ‘A’ of the current font. But the last one aligns the bottom edge of graphics on the baseline.

Other distinction between these commands consists in their work in the draft mode. The `\putimage` command does not test a required graphics file on existence and puts the filename at the center of bounding rectangle. But the `\includegraphics` searches the graphics file and puts the filename at the left top corner of bounding rectangle.

In version 1.04 of this package, we add a patch to the **graphics** package. The reason of this patch is the following. In latest versions of the **MiKTeX**, graphics rules are specified for many graphics formats. As a result, if a bounding box is not specified in parameters of the `\includegraphics` command, a file with `.bb` extension is tested on the ps-like bounding box specification. If such file does not exist, the `\includegraphics` command fails even if its parameters exactly specify the graphics width and height. We have changed the unconditional testing of the bounding box to the conditional one: testing is applied if the required file exists, otherwise, this operation is ignored.

2 The Implementation

Load required packages. We use some features of the `nccboxes` package in the `\putimage` command: the `\Strut` and `\jvbox` commands.

```

1 \*package
2 \RequirePackageWithOptions{graphicx}[1999/02/16]
3 \RequirePackage{nccboxes}[2002/01/09]

   Customize extension lists for dvips and dvipdf drivers.

4 \def\@tempa{dvips.def}
5 \ifx\Gin@driver\@tempa
6   \DeclareGraphicsExtensions{.eps,.ps,.eps.gz,.ps.gz,.eps.Z,%
7                               .bmp,.msp,.pcx,.pict,.pntg}
8 \else
9   \def\@tempa{dvipdf.def}
10  \ifx\Gin@driver\@tempa
11    \DeclareGraphicsExtensions{.eps,.ps,.eps.gz,.ps.gz,.eps.Z,%
12                                .bmp,.msp,.jpg}
13  \fi
14 \fi

```

Set the path list for search graphics files. It is not necessary to set the base directory in the list, because it is always searched first and the search in the graphics path list is applied only if the search in the base directory fails.

```

15 \graphicspath{{graphics/}}

```

`\draftgraphics` Specify graphics mode control commands:

```

\finalgraphics 16 \newcommand\draftgraphics{\Gin@drafttrue}
                17 \newcommand\finalgraphics{\Gin@draftfalse}

```

`\ipic` The `\ipic{<filename>}` command is equivalent to `\input{<filename>.pic}` command with search in the directory list specified by the `\graphicspath`. The file is opened within a group and the beginning and final spaces are removed in it.

```

18 \newcommand*\ipic[1]{%
19   \begingroup \let\input@path\Ginput@path
20   \ignorespaces\input{#1.pic}\unskip
21   \endgroup
22 }

```

`\putimage` Introduce the `\putimage` command. Using the empty definition we ensure the command was undefined before.

```

23 \newcommand{\putimage}{}
24 \def\putimage(#1,#2){%
25   \@ifnextchar[{\NCC@Gim(#1,#2)}{\NCC@Gim(#1,#2)[#1,#2]}%
26 }
27 \def\NCC@Gim(#1,#2)[#3,#4]{%
28   \ifGin@draft
29     \def\NCC@temp{##1,##2}##3{%
30       \edef\@tempa{##3}%

```

```

31     \put(0,0){%
32         \framebox{#1,#2}{\ttfamily\expandafter\strip@prefix\meaning\@tempa}%
33     }%
34 }%
35 \else
36     \def\NCC@temp{##1,##2}##3{%
37         \put{##1,##2}{%
38             \includegraphics[width=#3\unitlength,height=#4\unitlength]{##3}%
39         }%
40     }%
41 \fi
42 \@ifnextchar({\NCC@Gim@(#1,#2)}{\NCC@Gim@(#1,#2)(0,0)}{%
43 }
44 \def\NCC@Gim@(#1,#2)(#3,#4)#5{%
45     \jvbox{\Strut}[t]{%
46         \begin{picture}(#1,#2)\NCC@temp(#3,#4){#5}\end{picture}%
47     }%
48 }

```

`\Gin@setfile` The patch to the `\Gin@setfile` command slightly changes its behaviour: if bounding box calculations are required and the corresponding graphics rule contains an extension of file for reading the bounding box info, we skip the reading of given file if it does not exist.

```

49 \let\NCC@Ginsetfile\Gin@setfile
50 \def\Gin@setfile#1#2{%
51     \ifGin@bbox\else
52         \ifx\#2\\\else
53             \IfFileExists{\Gin@base#2}{\Gread@false}%
54         \fi
55     \fi
56     \NCC@Ginsetfile{#1}{#2}%
57 }
58 </package>

```