The gincltex Package

Martin Scharrer
martin@scharrer-online.de

http://www.ctan.org/pkg/gincltex

Version v0.3 – 2011/09/04

Abstract

This small package builds on the standard LaTeX package `graphicx` and allows external LaTeX source files to be included like graphic files, i.e. adds support for the '`.tex`' extension. Some of the lower level operations like clipping and trimming are implemented using the `pgf` package which supports both DVI/PS and PDF output. This package uses a very similar technique than the author's other package `adjustbox`, but provides a different interface.

Please note that this package is new and the implementation might change in future revisions. This might cause minor rounding differences in the exact size of the resulting TeX box around the included files.

1 Introduction

This small package builds on the standard LaTeX package `graphicx` and allows external LaTeX source files to be included like graphic files:

`\includegraphics`[⟨*options*⟩]{⟨*file*⟩`.tex`}

A LaTeX file included this way should result in an identical display as a tightly cropped EPS or PDF image of the same file (apart smaller rounding differences). Usually such files hold a picture environment like `picture`, `pspicture`, `pgfpicture` or `tikzpicture`, which may take advantage from the `standalone` class. In fact `gincltex` is used in newer versions of `standalone` to seamlessly switch between source and image files.

All options of `\includegraphics` described in the manual of `graphicx` (the `grfguide`) should be supported. Therefore it is possible to resize, rotate and clip the content of the LaTeX source file in the same way as for images.

An alternative is the `adjustbox` package from the same author which allows the same options as for `\includegraphics` for arbitrary TeX material:

`\adjustbox`{⟨*includegraphics options*⟩}{`\input`{⟨*file*⟩}}

2 Usage

After loading the package the '`.tex`' extension is supported by `\includegraphics` and the macro can be used in its normal form for LaTeX files. The content of the

file is typeset first inside an \hbox (the primitive version of \mbox) and then modified according to the given macro options. The graphicx package is automatically loaded.

2.1 Draft support

The package supports the draft option of graphicx which only displays an empty box with the file name for all included graphics. In this mode the source file should not be processed to reduce compile time. However the size of the resulting box from the source file must be know in order to reserve the required space. Therefore the *bounding box* information is cached for future runs with active draft option. The location where the information is cached can be controlled with the bb option.

2.2 Package options

draft
final

The draft and final options are directly passed to the loaded graphicx package. Having a different draft setting is not supported and the one used by the graphicx package will always take affect for gincltex.

bb

The place where the bounding box information is cached can be adjusted with the bb option. By default bb=aux is active which stores the bounding box information in the .aux file. With bb=file this information is written in EPS format into .tex.bb files, e.g. for each source file 'name.tex' a file 'name.tex.bb' is created.

3 Implementation

3.1 Package Option

At the moment the key=value format is simply hard coded.

```
\newif\if@gincltex@bbfile
\DeclareOption{bb=file}{\@gincltex@bbfiletrue}
\DeclareOption{bb=aux}{\@gincltex@bbfilefalse}
\DeclareOption{draft}{\PassOptionsToPackage{draft}{graphicx}}
\DeclareOption{draft=true}{\PassOptionsToPackage{draft}{graphix}}
\DeclareOption{draft=false}{\PassOptionsToPackage{final}{graphicx}}
\DeclareOption{final}{\PassOptionsToPackage{final}{graphicx}}
\DeclareOption{final=true}{\PassOptionsToPackage{final}{graphicx}}
\DeclareOption{final=false}{\PassOptionsToPackage{draft}{graphicx}}
\DeclareOption*{\PassOptionsToPackage\CurrentOption{adjustbox}}
\ProcessOptions*\relax
```

3.2 Requirements

The graphicx package is required. The pgf package is required for the graphic manipulations.

```
\RequirePackage{adjustbox}
```

3.3 Graphics Rule Macros

The following macro implement a *graphics rule* for LaTeX source code files.

\Gin@rule@.tex

This macro declares the graphics rule to the graphicx package.

```
\DeclareGraphicsRule{.tex}{tex}{.tex}{}
```

\gincltex@box

A savebox required to transfer material from the 'read' macro to the 'include' macro. Note that \@tempboxa is not used here because it might be used otherwise between the two macros.

```
\newsavebox\gincltex@box
```

\gincltex@boxfile

Macro to box the LaTeX source file. Because \includegraphics can be used inside this file certain internal graphicx macros must be reset to there default value. The argument is expanded first because it could include \Gin@ext. The content is stored with zero depth to achieve the same result as with included graphics.

```
\def\gincltex@boxfile#1{%
    \sbox\gincltex@box{{%
        \hbox{\vbox{%
            \hbox{%
                \edef\@tempa{{#1}}%
                \let\Gin@ext\relax
                \expandafter\input\@tempa
            }%
            \vskip\z@
        }}%
    }}%
}
```

\Ginclude@tex

This driver macro is called from the standard \includegraphics macro to include the LaTeX source file. Some \includegraphics options like angle are handled by wrapping this macro in the appropriate graphicx macro like \rotatebox, but others must be handled here.

```
\def\Ginclude@tex#1{%
    \begingroup
```

The content of the source file might have been already saved into the \gincltex@box by the \Gread@tex macro. If not it is saved here.

```
    \ifvoid\gincltex@box
        \gincltex@boxfile{#1}%
    \fi
```

The height, totalheight and width options are already processed and the final requested height and width to which the 'graphic' should be scaled to are provided. The internal form of the \resizebox macro is used for this.

```
    \resizebox*{\Gin@req@width}{\Gin@req@height}{%
        \ifGin@clip\expandafter\clipbox\else\/
            expandafter\clipbox\fi*{{\Gin@llx} {\/
            Gin@lly} {\Gin@urx} {\Gin@ury}}{\usebox\/
            gincltex@box}%
    }%
    \endgroup
}
```

The \Gread@tex macro is defined in two different ways depending how the bounding box information is preserved. This information is required to support the draft option of the graphicx package.

```
\if@gincltex@bbfile
```

Use a .tex.bb file to store the bounding box information. The standardised EPS format is used here, so that the \Gread@eps macro can be used.

An output register is required to write the .tex.bb files. Advanced users are allowed to predefine it manually in order to save a write register. Note that the writing is done inside the .aux file, therefore the \@mainaux handle could be used here, because it is closed while reading the .aux file.

```
\@ifundefined{gincltex@bbout}{\newwrite\/
    gincltex@bbout}{}
```

\gincltex@bb

Write the bounding box information to the .tex.bb file. The hi-resolution version is used to be more accurate. The code to write the normal version is disabled for now because it is unneeded and requires some non-trivial pgfmath calls.

Because this macro is executed inside the .aux file, which is read before the begin AND at the end of the document, the macro is defined as a no-op first two avoid unnecessary double execution.

```
\def\gincltex@bb#1#2#3#4#5{}
\AtBeginDocument{\let\gincltex@bb\gincltex@@bb}
\def\gincltex@@bb#1#2#3#4#5{%
    \begingroup
    \immediate\openout\gincltex@bbout=#1.bb\relax
    \immediate\write\gincltex@bbout{\@percentchar
        \@percentchar HiResBoundingBox: #2 #3 #4 #5}%
    \immediate\closeout\gincltex@bbout
    \endgroup
}
```

Storing the bounding box information in the `.aux` file.

```
\else
```

\Gread@tex@setbb

Auxiliary macro to set the bounding box macros.

```
\def\Gread@tex@setbb#1#2#3#4{%
    \def\Gin@llx{#1}%
    \def\Gin@lly{#2}%
    \def\Gin@urx{#3}%
    \def\Gin@ury{#4}%
}
```

\Gread@tex

Read the bounding box information. The only way to do this is to actually typeset the source file into a box. The box is then reused in the `\Ginclude@tex` macro, so there is no overhead. The bounding box information is written into the `.aux` file to avoid processing the source file in draft mode. However if the corresponding macro is not define yet (e.g. draft run without `.aux` file) the file must be read anyway.

```
\def\Gread@tex#1{%
    \ifcase0\ifGin@draft\@ifundefined{gincltex@bb@
       #1}{0}{1}\fi\relax
        \gincltex@boxfile{#1}%
        \def\Gin@llx{0}%
        \let\Gin@llx\Gin@lly
        \Gin@defaultbp\Gin@urx{\wd\gincltex@box}%
        \Gin@defaultbp\Gin@ury{\ht\gincltex@box}%
        \expandafter\xdef\csname gincltex@bb@#1\
           endcsname
            {{\Gin@llx}{\Gin@lly}{\Gin@urx}{\Gin@ury
               }}%
    \else
        \expandafter\expandafter\expandafter\
           Gread@tex@setbb
            \csname gincltex@bb@#1\endcsname
```

```
            \setbox\gincltex@box=\box\voidb@x
        \fi
        \if@filesw
            \immediate\write\@auxout{\string\gincltex@bb
                {#1}%
                 \csname gincltex@bb@#1\endcsname}%
        \fi
    }
```

`\gincltex@bb`

Simply define the corresponding bounding box macro.

```
    \def\gincltex@bb#1#2#3#4#5{%
        \global\@namedef{gincltex@bb@
            #1}{{#2}{#3}{#4}{#5}}%
    }

    \fi
    \endinput
```