The **figlatex** package[*]

Arnaud LEGRAND   Vincent DANJEAN

2021/01/03

Abstract

This package allows LaTeX to load XFig and SVG figures (or subfigures) with the \includegraphics command.

1   Introduction

1.1   General

\includegraphics   This package allows to load XFig figures from LaTeX sources with the \includegraphics command from the graphicx package:

    \includegraphics[width=\linewidth]{myfig.fig}

This package does not convert by itself XFig figures (see package texdepends for that). It only diverts the loading functions so that LaTeX includes the .pstex_t or the .pdftex_t.

1.2   Options

Available options for this package:

debug   **debug :** more messages written while compiling

figlatex.cfg   When the package is loaded, it tries to read a figlatex.cfg file. This file can
\debug   contains \debug to enable this option site wide.

1.3   Subfigures

Sometimes, several figures made with XFig should be similar. It is possible to make only one XFig figure and then create subfigures which contains only a subset of the depths of the original figure.

---

[*]This file has version number v2.4.2, last revised 2021/01/03.

2 Implementation

2.1 figlatex

1 ⟨∗package⟩

Options definitions

debug

```
2 \newif\ifFL@debug
3 \DeclareOption{debug}{%
4   \global\FL@debugtrue%
5 }
```

We load the config file if it exists. Commands for the config file are local.

```
6 {
```

\debug

```
7   \def\debug{%
8     \ExecuteOptions{debug}%
9   }

10   \InputIfFileExists{figlatex.cfg}{%
11     \typeout{****************************************^^J%
12             * Local config file figlatex.cfg used *^^J%
13             ****************************************}
14   }{%
15     \typeout{Found no configuration file for figlatex}
16   }
17 }
```

Run the options

```
18 \ProcessOptions
```

and requires some packages

```
19 \RequirePackage{ifthen}
20 \RequirePackage{ifpdf}
21 \RequirePackage{graphicx}
22 \RequirePackage{xstring}
```

XFig uses heavily the color package

```
23 \RequirePackage{color}
24 \RequirePackage{epstopdf}
```

\FL@dmessage   Write the message only if debug is set

```
25 \def\FL@dmessage#1{%
26   \ifFL@debug%
27     \message{#1}%
28   \fi%
29 }
```

\Ginclude@rawtex   `.pstex_t` or `.pdftex_t` are loaded with `\input`

```
30 \def\Ginclude@rawtex#1{%
31   \FL@dmessage{<#1>}%
32   \bgroup%
33   \def\@tempa{!}%
34   \ifx\Gin@scaley\@tempa%
35     \let\Gin@scaley\Gin@scalex%
36   \else%
37     \ifx\Gin@scalex\@tempa\let\Gin@scalex\Gin@scaley\fi%
38   \fi%
39   \hbox{%
40     \Gscale@box{\Gin@scalex}[\Gin@scaley]{%
41       \def\Gin@scalex{1}%
42       \let\Gin@scaley\Gin@exclamation% needed for pdflatex
43       \FL@dmessage{<loading rawtex #1>}%
44       \input{#1}}}%
45   \egroup}
```

\Ginclude@figtex   `.fig` are loaded with `\includegraphics` after modifying the name

```
46 \def\Ginclude@figtex#1{%
47   \message{<#1>}%
48   \bgroup%
49   \xdef\FL@filename{#1}%
50   \filename@parse{\FL@filename}%
51   \xdef\FL@filename{\filename@area\filename@base.\FL@figext}%
52   \FL@dmessage{<****loading fig \FL@filename>}%
53   \FL@includerawtexgraphics[]{\FL@filename}%
54   \egroup}
```

\Gread@figtex   `.fig` bounding box is the same as the corresponding `.pstex_t` or `.pdftex_t`

```
55 \newcommand{\Gread@figtex}[1]{%
56   %\typeout{Oh, file exists #1}%
57   \Gread@rawtex{#1}%
58 }
```

\Ginclude@svgtex   `.svg` are loaded with `\includegraphics` after modifying the name

```
59 \def\Ginclude@svgtex#1{%
60   \message{<#1>}%
61   \bgroup%
62   \xdef\FL@filename{#1}%
63   \filename@parse{\FL@filename}%
64   \xdef\FL@filename{\filename@area\filename@base.\FL@svgext}%
65   \FL@dmessage{<****loading svg \FL@filename>}%
66   \FL@includerawtexgraphics[]{\FL@filename}%
67   \egroup}
```

\Gread@svgtex   `.svg` bounding box is the same as the corresponding `.pstex_t` or `.pdftex_t`

```
68 \newcommand{\Gread@svgtex}[1]{%
69   %\typeout{Oh, file exists #1}%
```

```
70    \Gread@rawtex{#1}%
71 }
```

Declare the right things depending whether we build a DVI or a PDF file.

```
72 \ifpdf
```

PDF part

```
73    \def\FL@figext{pdftex_t}%
74    \def\FL@svgext{pdftex_t}%
```

To read bounding box information:

```
75    \let\Gread@rawtex\Gread@pdf%
76    \DeclareGraphicsRule{.pdftex}{pdf}{*}{}%
77    \DeclareGraphicsRule{.pdftex_t}{rawtex}{.pdftex}{}%
78    \DeclareGraphicsRule{.fig}{figtex}{.pdftex}{}%
79    \DeclareGraphicsRule{.svg}{svgtex}{.pdftex}{}%
80 \else
```

DVI part

```
81    \def\FL@figext{pstex_t}%
82    \def\FL@svgext{pstex_t}%
```

To read bounding box information:

```
83    \let\Gread@rawtex\Gread@eps%
84    \DeclareGraphicsRule{.pstex}{eps}{*}{}%
85    \DeclareGraphicsRule{.pstex_t}{rawtex}{.pstex}{}%
86    \DeclareGraphicsRule{.fig}{figtex}{.pstex}{}%
87    \DeclareGraphicsRule{.svg}{svgtex}{.pstex}{}%
88 \fi
```

This command is not really necessary here (code could be added in `\Ginclude@figtex`). However, it's very convinient for the `texdepends` package.

```
89 \newcommand{\FL@includerawtexgraphics}[2][]{%
90    \includegraphics[#1]{#2}%
91 }
```

Addition of the `subfig=name` key for the options of the `\includegraphicx[..]{..}` macro.

```
92 \def\FL@exclamation{!}%
93 \let\FL@subfig@name\FL@exclamation%
94 \define@key{Gin}{subfig}{%
95    \typeout{using subfig #1}%
96    \def\FL@subfig@name{#1}%
97 }
98 \def\FL@sepdefault{_}%
```

We should ckeck if the #1 file exists (XXX.subfig) This is not yet done but texdepends divert this macro to record the dependency

```
99 \def\FL@subfig@check#1{%
100 }
```

We divert `\Gin@getbase` to modify the file name if the `subfig` key is used.

```
101 \def\FL@Gin@getbase#1{%
102 \typeout{In getbase #1 with \FL@subfig@name / \FL@exclamation}%
103 \ifx\FL@subfig@name\FL@exclamation%
104 \else%
105 \typeout{getbase lets go (\filename@area)(\filename@base)(#1)}%
106 \IfStrEq*{#1}{.fig}{%
107   % If we're including a subfig
108   \IfFileExists{\filename@area\filename@base#1}{%
109     % If the original .fig file exists, just change the name so that
110     % it comprises the full path (file has been looked into
111     % texgraphics, etc.)
112     % Warning: the found filename is quoted :-(
113     \StrBetween[1,2]{\@filef@und}{"}{"}[\FL@filename@unquoted]
114     \filename@parse{\FL@filename@unquoted}%
115     % Check if the subfig file exists
116     \FL@subfig@check{\filename@area\filename@base.subfig}%
117   }{%
118     % Else, write a warning (LaTeX will warn as well but it doesn't
119     % hurt)
120     \PackageWarning{figlatex}{No file
121       '\filename@area\filename@base#1'}%
122   }
123   % Add the subfig suffix to the fig name.
124   \edef\filename@base{\filename@base\FL@sepdefault\FL@subfig@name}%
125 }{%
126 }%
127 \IfStrEq*{#1}{.svg}{%
128   % If we're including a subfig
129   \IfFileExists{\filename@area\filename@base#1}{%
130     % If the original .svg file exists, just change the name so that
131     % it comprises the full path (file has been looked into
132     % texgraphics, etc.)
133     % Warning: the found filename is quoted :-(
134     \StrBetween[1,2]{\@filef@und}{"}{"}[\FL@filename@unquoted]
135     \filename@parse{\FL@filename@unquoted}%
136     % Check if the subfig file exists
137     \FL@subfig@check{\filename@area\filename@base.subfig}%
138   }{%
139     % Else, write a warning (LaTeX will warn as well but it doesn't
140     % hurt)
141     \PackageWarning{figlatex}{No file
142       '\filename@area\filename@base#1'}%
143   }
144   % Add the subfig suffix to the svg name.
145   \edef\filename@base{\filename@base\FL@sepdefault\FL@subfig@name}%
146 }{%
147 }%
148 \fi%
```

```
149    \FL@orig@Gin@getbase{#1}%
150 }
151 \let\FL@orig@Gin@getbase\Gin@getbase
152 \let\Gin@getbase\FL@Gin@getbase
153
154 %\@namedef{Gin@rule@.fig}#1{%
155
156 %}
157 ⟨/package⟩
```

The sample config file `figlatex.cfg`...

```
158 ⟨*config⟩

159    %\debug

160 ⟨/config⟩
```

2.2 texgraphicx

```
161 ⟨*compat⟩
```

`texgraphicx.sty` was only a wrapper arround `figlatex.sty`. Now, an error is generated.

```
162 \PackageError{texgraphicx}{'texgraphicx' is now
163    gone\MessageBreak%
164    Please, switching to 'figlatex'
165    package\MessageBreak}{}
166 \ProcessOptions
167 ⟨/compat⟩
```

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