The grfext package

Heiko Oberdiek*

2019/12/03 v1.3

Abstract

This package provides macros for adding and reordering graphics extensions of package graphics.

Contents

*Please report any issues at https://github.com/ho-tex/grfext/issues

1 Documentation

1.1 Introduction

If you are not familiar with LaTeX's graphics bundle, please read its documentation `grffile` [1]. The bundle contains two packages for graphics inclusion: graphics and graphicx. The first one is loaded by the second one that adds a key value interface.

Graphics files are included in both cases by macro `\includegraphics`. The file name extension can be omitted. Then the graphics package goes through a list of known extensions until it finds the graphics file. This extension list is set by `\DeclareGraphicsExtensions`. The previous contents of the list is overwritten.

1.2 User interface

This package grfext provides macros that adds entries to the list or remove them. The list may be empty or even undefined before. It is always defined afterwards, but can be empty (especially after removing entries).

`\AppendGraphicsExtensions` * {⟨*ext-list*⟩}
`\PrependGraphicsExtensions` * {⟨*ext-list*⟩}

The argument ⟨*ext-list*⟩ is a comma separated list whose entries are file name extensions including the dot. But first the entries are removed from graphics' extension list to avoid multiple occurences of the same extension.

Then macro `\AppendGraphicsExtensions` adds the entries after the end of graphics' list, whereas macro `\PrependGraphicsExtensions` puts them in front of the list. The order matters if a graphics file is available in different acceptable formats. Then the first extension wins.

The star version of these commands only adds an extensions, if a specific graphics rule exists for that extension.

`\RemoveGraphicsExtensions` {⟨*ext-list*⟩}

All occurences of file extensions in ⟨*ext-list*⟩ are removed from graphics' extension list.

1.3 Package loading

The package does not define any options. It is loaded as usual in LaTeX, e.g.:

```
\usepackage{grfext}
```

`\PrintGraphicsExtensions`

Macro `\PrintGraphicsExtensions` writes the current graphics extensions list in the `.log` file. The macros described before do this automatically after their operation.

1.4 Option support for package graphicx

Package graphicx uses the interface of package keyval in order to specify options for `\includegraphics`. The options can also be set using

```
\setkeys{Gin}{⟨options⟩}
```

The four user macros with the two star forms are available as options in family `Gin` as well:

```
AppendGraphicsExtensions={⟨ext-list⟩}
AppendGraphicsExtensions*={⟨ext-list⟩}
PrependGraphicsExtensions={⟨ext-list⟩}
PrependGraphicsExtensions*{⟨ext-list⟩}
RemoveGraphicsExtensions={⟨ext-list⟩}
PrintGraphicsExtensions
```

This makes it easier to locally change the extension list for an included graphics, e.g.:

```
\includegraphics[RemoveGraphicsExtensions={.pdf,PDF}]{image}
```

1.5 plain TEX

LATEX's graphics packages can also be used with plain TEX. The necessary basic LATEX macros are defined in `miniltx.tex`. This package grfext also relies on it. Example:

```
\input miniltx.tex\relax
\def\Gin@driver{pdftex.def}
\input graphicx.sty\relax
\input grfext.sty\relax
\resetatcatcode
```

2 Implementation

1 ⟨*package⟩

2.1 Reload check and identification

Reload check, especially if the package is not used with LATEX.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3   \catcode13=5 % ^^M
4   \endlinechar=13 %
5   \catcode35=6 % #
6   \catcode39=12 % '
7   \catcode44=12 % ,
8   \catcode45=12 % -
9   \catcode46=12 % .
10  \catcode58=12 % :
11  \catcode64=11 % @
12  \catcode123=1 % {
13  \catcode125=2 % }
14  \expandafter\let\expandafter\x\csname ver@grfext.sty\endcsname
15  \ifx\x\relax % plain-TeX, first loading
16  \else
17    \def\empty{}%
18    \ifx\x\empty % LaTeX, first loading,
19      % variable is initialized, but \ProvidesPackage not yet seen
20    \else
21      \expandafter\ifx\csname PackageInfo\endcsname\relax
22        \def\x#1#2{%
23          \immediate\write-1{Package #1 Info: #2.}%
```

```
        }%
      \else
        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
      \fi
      \x{grfext}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@grfext.sty\endcsname
\ProvidesPackage{grfext}%
  [2019/12/03 v1.3 Manage graphics extensions (HO)]%
```

2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname grfext@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
```

```
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\grfext@AtEnd{%
    \grfext@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{58}{12}% :
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\edef\grfext@AtEnd{\grfext@AtEnd\noexpand\endinput}
```

2.3 plain TEX

\@expandtwoargs Requirement is `miniltx.tex`, but we need also LATEX's `\@expandtwoargs`.

```
\@ifundefined{@expandtwoargs}{%
  \def\@expandtwoargs#1#2#3{%
    \edef\reserved@a{\noexpand#1{#2}{#3}}%
    \reserved@a
  }%
}{}
```

2.4 Add

\AppendGraphicsExtensions

```
\newcommand*{\AppendGraphicsExtensions}{%
  \@ifundefined{Gin@extensions}{%
    \let\Gin@extensions\@empty
  }{}%
  \@ifstar{\grfext@Append\grfext@Check}{\grfext@Append\grfext@@Add}%
}%
```

\grfext@Append

```
\def\grfext@Append#1#2{%
  \let\grfext@Print\@gobble
  \edef\grfext@next{%
    \noexpand\grfext@Add\noexpand#1{%
```

```
      \zap@space#2 \@empty
    }{\noexpand\Gin@extensions,}{}%
  }%
  \grfext@next
  \let\grfext@Print\grfext@@Print
  \grfext@Print\AppendGraphicsExtensions
}
```

`\PrependGraphicsExtensions`

```
\newcommand*{\PrependGraphicsExtensions}{%
  \@ifundefined{Gin@extensions}{%
    \let\Gin@extensions\@empty
  }{}%
  \@ifstar{\grfext@Prepend\grfext@Check}{\grfext@Prepend\grfext@@Add}%
}%
```

`\grfext@Prepend`

```
\def\grfext@Prepend#1#2{%
  \let\grfext@Print\@gobble
  \edef\grfext@next{%
    \noexpand\grfext@Add\noexpand#1{%
      \zap@space#2 \@empty
    }{}{,\noexpand\Gin@extensions}%
  }%
  \grfext@next
  \let\grfext@Print\grfext@@Print
  \grfext@Print\PrependGraphicsExtensions
}
```

`\grfext@Add`

```
\def\grfext@Add#1#2{%
  #1{#2}%
}
```

`\grfext@@Add`

```
\def\grfext@@Add#1#2#3{%
  \RemoveGraphicsExtensions{#1}%
  \ifx\Gin@extensions\@empty
    \def\Gin@extensions{#1}%
  \else
    \edef\Gin@extensions{#2#1#3}%
  \fi
}
```

2.5 Check

`\grfext@Check`

```
\def\grfext@Check#1{%
  \let\grfext@tmp\@empty
  \@for\grfext@ext:=#1\do{%
    \@ifundefined{Gin@rule@\grfext@ext}{%
    }{%
      \ifx\grfext@tmp\@empty
        \let\grfext@tmp\grfext@ext
      \else
        \edef\grfext@tmp{\grfext@tmp,\grfext@ext}%
      \fi
```

```
    }%
  }%
  \ifx\grfext@tmp\@empty
    \def\grfext@next##1##2{}%
  \else
    \edef\grfext@next{%
      \noexpand\grfext@@Add{\grfext@tmp}%
    }%
  \fi
  \grfext@next
}
```

2.6 Remove

\RemoveGraphicsExtensions

```
\newcommand*{\RemoveGraphicsExtensions}[1]{%
  \@ifundefined{Gin@extensions}{%
    \def\Gin@extensions{}%
  }{%
    \edef\grfext@tmp{\zap@space#1 \@empty}%
    \@for\grfext@ext:=\grfext@tmp\do{%
      \def\grfext@next{%
        \let\grfext@tmp\Gin@extensions
        \@expandtwoargs
        \@removeelement\grfext@ext\Gin@extensions\Gin@extensions
        \ifx\grfext@tmp\Gin@extensions
          \let\grfext@next\relax
        \fi
        \grfext@next
      }%
      \grfext@next
    }%
  }%
  \grfext@Print\RemoveGraphicsExtensions
}
```

2.7 Print

```
\RequirePackage{infwarerr}[2007/09/09]
```

\PrintGraphicsExtensions

```
\def\PrintGraphicsExtensions{%
  \grfext@Print\PrintGraphicsExtensions
}
```

\grfext@Print

```
\def\grfext@Print#1{%
  \@PackageInfo{grfext}{%
    Graphics extension search list:\MessageBreak
    \@ifundefined{Gin@extensions}{%
      <unavailable>%
    }{%
      [\Gin@extensions]%
    }\MessageBreak
    \string#1%
  }%
}
```

\grfext@@Print

```
\let\grfext@@Print\grfext@Print
```

2.8 Defining options for package graphicx

```
\RequirePackage{kvdefinekeys}[2010/03/01]
\kv@define@key{Gin}{AppendGraphicsExtensions}{%
  \AppendGraphicsExtensions{#1}%
}
\kv@define@key{Gin}{AppendGraphicsExtensions*}{%
  \AppendGraphicsExtensions*{#1}%
}
\kv@define@key{Gin}{PrependGraphicsExtensions}{%
  \PrependGraphicsExtensions{#1}%
}
\kv@define@key{Gin}{PrependGraphicsExtensions*}{%
  \PrependGraphicsExtensions*{#1}%
}
\kv@define@key{Gin}{RemoveGraphicsExtensions}{%
  \RemoveGraphicsExtensions{#1}%
}
\kv@define@key{Gin}{PrintGraphicsExtensions}[]{%
  \PrintGraphicsExtensions
}
```

```
\grfext@AtEnd%
```
⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/grfext/grfext.dtx The source file.

CTAN:macros/latex/contrib/grfext/grfext.pdf Documentation.

Bundle. All the packages of the bundle 'grfext' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/grfext.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `grfext.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip grfext.tds.zip -d ~/texmf
```

[1] CTAN:pkg/grfext

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain TEX:

```
tex grfext.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
grfext.sty → tex/latex/grfext/grfext.sty
grfext.pdf → doc/latex/grfext/grfext.pdf
grfext.dtx → source/latex/grfext/grfext.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

3.4 Refresh file name databases

If your TEX distribution (TEX Live, mikTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{grfext.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLATEX:

```
pdflatex grfext.dtx
makeindex -s gind.ist grfext.idx
pdflatex grfext.dtx
makeindex -s gind.ist grfext.idx
pdflatex grfext.dtx
```

4 References

[1] David Carlisle, Sebastian Rahtz: *The graphics package*; 2006/02/20 v1.0o; CTAN:macros/latex/required/graphics/graphics.dtx.

5 History

[2007/09/30 v1.0]

- First public version.

[2010/08/19 v1.1]

- User macros are also made available as keyval options for package graphicx.

[2016/05/16 v1.2]

- Documentation updates.

[2019/12/03 v1.3]

- Documentation updates.

6 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; plain numbers refer to the code lines where the entry is used.