The grfext package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/09/30 v1.0

Abstract

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

Contents

1 Documentation **2**
1.1 Introduction . . . 2
1.2 User interface . . . 2
1.3 Package loading . . . 2
1.4 plain-TEX . . . 2

2 Implementation **3**
2.1 Reload check and identification . . . 3
2.2 Catcodes . . . 4
2.3 plain-TEX . . . 4
2.4 Add . . . 4
2.5 Check . . . 5
2.6 Remove . . . 6
2.7 Print . . . 6

3 Test **7**
3.1 Catcode checks for loading . . . 7
3.2 Macro tests . . . 8
3.2.1 Preamble . . . 8
3.2.2 Detection of unwanted space . . . 8
3.2.3 Generic test macros . . . 8
3.2.4 Test for \AppendGraphicsExtensions . . . 9
3.2.5 Test for \PrependGraphicsExtensions . . . 9
3.2.6 Test for \RemoveGraphicsExtensions . . . 10

4 Installation **11**
4.1 Download . . . 11
4.2 Bundle installation . . . 11
4.3 Package installation . . . 11
4.4 Refresh file name databases . . . 11
4.5 Some details for the interested . . . 12

5 References **12**

6 History **12**
[2007/09/30 v1.0] . . . 12

7 Index **12**

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 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

```
⟨*package⟩
```

2.1 Relead check and identification

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

```
\begingroup
  \catcode44 12 % ,
  \catcode45 12 % -
  \catcode46 12 % .
  \catcode58 12 % :
  \catcode64 11 % @
  \expandafter\let\expandafter\x\csname ver@grfext.sty\endcsname
  \ifcase 0%
    \ifx\x\relax % plain
    \else
      \ifx\x\empty % LaTeX
      \else
        1%
      \fi
    \fi
  \else
    \catcode35 6 % #
    \catcode123 1 % {
    \catcode125 2 % }
    \expandafter\ifx\csname PackageInfo\endcsname\relax
      \def\x#1#2{%
        \immediate\write-1{Package #1 Info: #2.}%
      }%
    \else
      \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
    \fi
    \x{grfext}{The package is already loaded}%
    \endgroup
    \expandafter\endinput
  \fi
\endgroup
```

Package identification:

```
\begingroup
  \catcode35 6 % #
  \catcode40 12 % (
  \catcode41 12 % )
  \catcode44 12 % ,
  \catcode45 12 % -
  \catcode46 12 % .
  \catcode47 12 % /
  \catcode58 12 % :
  \catcode64 11 % @
  \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\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@grfext.sty\endcsname
\ProvidesPackage{grfext}%
  [2007/09/30 v1.0 Managing graphics extensions (HO)]
```

2.2 Catcodes

```
\begingroup
  \catcode123 1 % {
  \catcode125 2 % }
  \def\x{\endgroup
    \expandafter\edef\csname grfext@AtEnd\endcsname{%
      \catcode35 \the\catcode35\relax
      \catcode64 \the\catcode64\relax
      \catcode123 \the\catcode123\relax
      \catcode125 \the\catcode125\relax
    }%
  }%
\x
\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{61}{12}% =
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
```

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

\grfext@AtEnd
⟨/package⟩
```

3 Test

Load `miniltx` for standard catcode test.

```
⟨*test1⟩
\input miniltx.tex\relax
\resetatcatcode
⟨/test1⟩
```

3.1 Catcode checks for loading

```
⟨*test1⟩

\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

```

```
\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input grfext.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\{=1 %
  \catcode`\}=2 %
  \catcode`\#=6 %
  \catcode`\[=12 %
  \catcode`\]=12 %
  \catcode`\%=14 %
  \catcode`\ =10 %
  \catcode13=5 %
  \LoadCommand
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 Macro tests

3.2.1 Preamble

⟨∗test2⟩

```
\NeedsTeXFormat{LaTeX2e}
\nofiles
\documentclass{article}
\usepackage[pdftex]{graphics}
\usepackage{grfext}[2007/09/30]
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\makeatletter
```

3.2.2 Detection of unwanted space

```
\let\orig@qstest\qstest
\let\orig@endqstest\endqstest
\def\qstest#1#2{%
  \orig@qstest{#1}{#2}%
  \setbox0\hbox\bgroup\begingroup\ignorespaces
}
\def\endqstest{%
  \endgroup\egroup
  \Expect*{\the\wd0}{0.0pt}%
  \orig@endqstest
}
```

3.2.3 Generic test macros

```
\def\Test{\@ifnextchar\bgroup\TestInitList\TestInitCmd}%
\def\TestInitList#1{%
  \def\Gin@extensions{#1}%
  \@Test
}
\def\TestInitCmd#1{%
  \let\Gin@extensions#1\relax
  \@Test
}
\def\@Test#1#2{%
  \TestCmd{#1}%
  \expandafter\Expect\expandafter*{\Gin@extensions}{#2}%
}
```

3.2.4 Test for \AppendGraphicsExtensions

```
\begin{qstest}{AppendGraphicsExtensions}{AppendGraphicsExtensions}
  \let\TestCmd\AppendGraphicsExtensions
  \Test\@undefined{.test}{.test}%
  \Test\relax{.test}{.test}%
  \Test{}{.test}{.test}%
  \Test{.foo}{.test}{.foo,.test}%
  \Test{.foo,.bar}{.test}{.foo,.bar,.test}%
  \Test{.test}{.test}{.test}%
  \Test{.foo,.test}{.test}{.foo,.test}%
  \Test{.test,.foo}{.test}{.foo,.test}%
  \Test{.foo,.test,.bar}{.test}{.foo,.bar,.test}%
  \Test{.foo}{.test,.bar}{.foo,.test,.bar}%
  \Test{.foo}{.test,.foo}{.test,.foo}%
  \Test{.foo}{ .test ,.bar , .x, .y }{.foo,.test,.bar,.x,.y}%
  \def\TestCmd{\AppendGraphicsExtensions*}%
  \Test\@undefined{.test}{}%
  \Test\relax{.test}{}%
  \Test{}{.test}{}%
  \Test{.test}{.test}{.test}%
  \Test{.foo}{.test}{.foo}%
  \Test{.foo,.bar}{.test}{.foo,.bar}%
  \Test{.test}{.test}{.test}%
  \Test{.foo,.test}{.test}{.foo,.test}%
  \Test{.test,.foo}{.test}{.test,.foo}%
  \Test{.foo,.test,.bar}{.test}{.foo,.test,.bar}%
  \Test{.foo}{.test,.bar}{.foo}%
  \Test{.foo}{.test,.foo}{.foo}%
  \DeclareGraphicsRule{.test}{test}{.test}{}%
  \Test\@undefined{.test}{.test}%
  \Test\relax{.test}{.test}%
  \Test{}{.test}{.test}%
  \Test{.foo}{.test}{.foo,.test}%
  \Test{.foo,.bar}{.test}{.foo,.bar,.test}%
  \Test{.test}{.test}{.test}%
  \Test{.foo,.test}{.test}{.foo,.test}%
  \Test{.test,.foo}{.test}{.foo,.test}%
  \Test{.foo,.test,.bar}{.test}{.foo,.bar,.test}%
  \Test{.foo}{.test,.bar}{.foo,.test}%
  \Test{.foo}{.test,.foo}{.foo,.test}%
  \DeclareGraphicsRule{.bar}{bar}{.bar}{}%
  \Test{.foo}{.test,.bar}{.foo,.test,.bar}%
  \Test{.bar}{.test,.bar}{.test,.bar}%
  \Test{.bar}{.bar,.test}{.bar,.test}%
\end{qstest}
```

3.2.5 Test for \PrependGraphicsExtensions

```
\begin{qstest}{PrependGraphicsExtensions}{PrependGraphicsExtensions}
  \let\TestCmd\PrependGraphicsExtensions
```

```
  \Test\@undefined{.test}{.test}%
  \Test\relax{.test}{.test}%
  \Test{}{.test}{.test}%
  \Test{.foo}{.test}{.test,.foo}%
  \Test{.foo,.bar}{.test}{.test,.foo,.bar}%
  \Test{.test}{.test}{.test}%
  \Test{.foo,.test}{.test}{.test,.foo}%
  \Test{.test,.foo}{.test}{.test,.foo}%
  \Test{.foo,.test,.bar}{.test}{.test,.foo,.bar}%
  \Test{.foo}{.test,.bar}{.test,.bar,.foo}%
  \Test{.foo}{.test,.foo}{.test,.foo}%
  \Test{.foo}{ .test ,.bar , .x, .y }{.test,.bar,.x,.y,.foo}%
  \def\TestCmd{\PrependGraphicsExtensions*}%
  \Test\@undefined{.test}{}%
  \Test\relax{.test}{}%
  \Test{}{.test}{}%
  \Test{.test}{.test}{.test}%
  \Test{.foo}{.test}{.foo}%
  \Test{.foo,.bar}{.test}{.foo,.bar}%
  \Test{.test}{.test}{.test}%
  \Test{.foo,.test}{.test}{.foo,.test}%
  \Test{.test,.foo}{.test}{.test,.foo}%
  \Test{.foo,.test,.bar}{.test}{.foo,.test,.bar}%
  \DeclareGraphicsRule{.test}{test}{.test}{}%
  \Test\@undefined{.test}{.test}%
  \Test\relax{.test}{.test}%
  \Test{}{.test}{.test}%
  \Test{.foo}{.test}{.test,.foo}%
  \Test{.foo,.bar}{.test}{.test,.foo,.bar}%
  \Test{.test}{.test}{.test}%
  \Test{.foo,.test}{.test}{.test,.foo}%
  \Test{.test,.foo}{.test}{.test,.foo}%
  \Test{.foo,.test,.bar}{.test}{.test,.foo,.bar}%
  \Test{.foo}{.test,.bar}{.test,.foo}%
  \Test{.foo}{.test,.foo}{.test,.foo}%
  \DeclareGraphicsRule{.bar}{bar}{.bar}{}%
  \Test{.foo}{.test,.bar}{.test,.bar,.foo}%
  \Test{.bar}{.test,.bar}{.test,.bar}%
  \Test{.bar}{.bar,.test}{.bar,.test}%
\end{qstest}
```

3.2.6 Test for \RemoveGraphicsExtensions

```
\begin{qstest}{RemoveGraphicsExtensions}{RemoveGraphicsExtensions}
  \let\TestCmd\RemoveGraphicsExtensions
  \Test\@undefined{.test}{}%
  \Test\relax{.test}{}%
  \Test{}{.test}{}%
  \Test{.foo}{.test}{.foo}%
  \Test{.foo,.bar}{.test}{.foo,.bar}%
  \Test{.test}{.test}{}%
  \Test{.foo,.test}{.test}{.foo}%
  \Test{.test,.foo}{.test}{.foo}%
  \Test{.foo,.test,.bar}{.test}{.foo,.bar}%
  \Test{.test,.test}{.test}{}%
  \Test{.foo,.test,.bar,.test,.test,.bar}{.test}{.foo,.bar,.bar}%
  \Test{.foo,.test,.bar,.foo,.test,.bar}{.test,.bar}{.foo,.foo}%
  \Test{.foo,.test,.bar,.foo,.test,.bar}{ .test , .bar }{.foo,.foo}%
\end{qstest}
```

```
\begin{document}
\end{document}
```
⟨/test2⟩

4 Installation

4.1 Download

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

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

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

Bundle. All the packages of the bundle 'oberdiek' 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/oberdiek.tds.zip

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

4.2 Bundle installation

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

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

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package attachfile2 comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 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/oberdiek/grfext.sty
grfext.pdf            → doc/latex/oberdiek/grfext.pdf
test/grfext-test1.tex → doc/latex/oberdiek/test/grfext-test1.tex
test/grfext-test2.tex → doc/latex/oberdiek/test/grfext-test2.tex
grfext.dtx            → source/latex/oberdiek/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.

4.4 Refresh file name databases

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

[1] ftp://ftp.ctan.org/tex-archive/

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is pdftk, e.g. unpack the file into the current directory:

```
pdftk grfext.pdf unpack_files output .
```

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
```

5 References

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

6 History

[2007/09/30 v1.0]

- First public version,

7 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.

Symbols

\# 210, 272
\% 275
\@ 211, 268
\@PackageInfo 190
\@Test 309, 313, 315
\@empty 101, 109, 118, 126, 138, 145, 149, 156, 169
\@expandtwoargs 93, 173
\@firstofone 219, 222
\@for 146, 170
\@gobble 106, 123, 216, 224
\@ifnextchar 306
\@ifstar 103, 120

\@ifundefined 93, 100, 117, 147, 166, 192
\@removeelement 174
\@undefined 321, 334, 347, 365, 378, 389, 407
\[. 273
\\ 269
\{ 208, 270
\} 209, 271
\] 274

\␣ 276

A

\advance 249, 257
\AppendGraphicsExtensions 2, 99, 114, 320, 333

B

\begin 319, 363, 405, 421
\body 228, 232

C

\catcode . . . 3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 62, 63, 66, 67, 68, 69, 73, 74, 75, 76, 80, 82, 208, 209, 210, 211, 246, 255, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277
\count@ 213, 242, 246, 248, 249, 253, 255, 256, 257
\countdef 213
\csname 8, 21, 45, 58, 65, 212, 215, 218, 221, 260, 282

D

\DeclareGraphicsRule 346, 358, 388, 400
\do 146, 170
\documentclass 288

E

\empty 12
\end 283, 362, 404, 420, 422
\endcsname 8, 21, 45, 58, 65, 212, 215, 218, 221, 260, 282
\endinput 30
\endqstest 296, 301
\Expect 303, 317

G

\Gin@extensions 101, 110, 118, 127, 138, 139, 141, 167, 172, 174, 175, 195, 308, 312, 317
\grfext@@Add 103, 120, 136, 160
\grfext@@Print 113, 130, 200
\grfext@Add 108, 125, 133
\grfext@Append 103, 105
\grfext@AtEnd 78, 79, 201
\grfext@Check 103, 120, 144
\grfext@ext 146, 147, 150, 152, 170, 174
\grfext@next . . . 107, 112, 124, 129, 157, 159, 163, 171, 176, 178, 180
\grfext@Prepend 120, 122
\grfext@Print 106, 113, 114, 123, 130, 131, 183, 187, 189, 200
\grfext@tmp 145, 149, 150, 152, 156, 160, 169, 170, 172, 175

H

\hbox 299

I

\ifcase 9
\ifnum 248, 256
\ifx . . . 10, 12, 21, 45, 53, 138, 149, 156, 175, 212, 215, 218, 221, 260
\ignorespaces 299
\immediate 23, 47
\IncludeTests 292
\input 204, 261
\iterate 229, 231, 233

L

\LoadCommand 261, 278
\LogTests 293
\loop 227, 243, 254

M

\makeatletter 294
\MessageBreak 191, 196

N

\NeedsTeXFormat 286
\newcommand 99, 116, 165
\next 233, 235, 237
\nofiles 287

O

\orig@endqstest 296, 304
\orig@qstest 295, 298

P

\PackageInfo 26
\PrependGraphicsExtensions 116, 131, 364, 377
\PrintGraphicsExtensions 2, 186
\ProvidesPackage 59

Q

\qstest 295, 297

R

\RangeCatcodeInvalid 252, 264, 265, 266, 267
\RemoveGraphicsExtensions 2, 137, 165, 406
\repeat 227, 239, 250, 258
\RequirePackage 185
\reserved@a 95, 96
\resetatcatcode 205
\RestoreCatcodes . . 241, 244, 245, 279

S

\setbox 299

T

\Test 263, 281, 306, 321, 322, 323, 324, 325, 326, 327, 328,

329, 330, 331, 332, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 359, 360, 361, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 401, 402, 403, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419
\TestCmd .. 316, 320, 333, 364, 377, 406
\TestInitCmd 306, 311
\TestInitList 306, 307
\the 66, 67, 68, 69, 80, 246, 303
\TMP@EnsureCode 77, 84, 85, 86, 87, 88, 89, 90, 91, 92

U

\usepackage 289, 290, 291

W

\wd 303
\write 23, 47

X

\x 8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72

Z

\zap@space 109, 126, 169