

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- <code>T_EX</code>	2
2	Implementation	3
2.1	Relead check and identification	3
2.2	Catcodes	4
2.3	plain- <code>T_EX</code>	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 <code>\AppendGraphicsExtensions</code>	9
3.2.5	Test for <code>\PrependGraphicsExtensions</code>	9
3.2.6	Test for <code>\RemoveGraphicsExtensions</code>	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 L^AT_EX'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 * {\langle ext-list \rangle}
\PrependGraphicsExtensions * {\langle ext-list \rangle}
```

The argument `\langle ext-list \rangle` 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 occurrences 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 {\langle ext-list \rangle}
```

All occurrences of file extensions in `\langle ext-list \rangle` are removed from `graphics`' extension list.

1.3 Package loading

The package does not define any options. It is loaded as usual in L^AT_EX, 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-T_EX

L^AT_EX's graphics packages can also be used with plain-T_EX. The necessary basic L^AT_EX 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 Reread check and identification

Reread check, especially if the package is not used with L^AT_EX.

```

2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@grfext.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{grfext}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup

```

Package identification:

```

33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46 \def\x#1#2#3[#4]{\endgroup
47 \immediate\write-1{Package: #3 #4}%
48 \xdef#1{#4}%
49 }%
50 \else
51 \def\x#1#2[#3]{\endgroup
52 #2[#3]}%

```

```

53      \ifx#1\relax
54      \xdef#1{#3}%
55      \fi
56    }%
57  \fi
58 \expandafter\x\csname ver@grfext.sty\endcsname
59 \ProvidesPackage{grfext}%
60 [2007/09/30 v1.0 Managing graphics extensions (H0)]

```

2.2 Catcodes

```

61 \begingroup
62   \catcode123 1 % {
63   \catcode125 2 % }
64   \def\x{\endgroup
65     \expandafter\edef\csname grfext@AtEnd\endcsname{%
66       \catcode35 \the\catcode35\relax
67       \catcode64 \the\catcode64\relax
68       \catcode123 \the\catcode123\relax
69       \catcode125 \the\catcode125\relax
70     }%
71   }%
72 \x
73 \catcode35 6 % #
74 \catcode64 11 % @
75 \catcode123 1 % {
76 \catcode125 2 % }
77 \def\TMP@EnsureCode#1#2{%
78   \edef\grfext@AtEnd{%
79     \grfext@AtEnd
80     \catcode#1 \the\catcode#1\relax
81   }%
82   \catcode#1 #2\relax
83 }
84 \TMP@EnsureCode{42}{12}% *
85 \TMP@EnsureCode{44}{12}% ,
86 \TMP@EnsureCode{47}{12}% /
87 \TMP@EnsureCode{58}{12}% :
88 \TMP@EnsureCode{60}{12}% <
89 \TMP@EnsureCode{61}{12}% =
90 \TMP@EnsureCode{62}{12}% >
91 \TMP@EnsureCode{91}{12}% [
92 \TMP@EnsureCode{93}{12}% ]

```

2.3 plain-TeX

\@expandtwoargs Requirement is minilTeX, but we need also L^AT_EX's \@expandtwoargs.

```

93 \@ifundefined{@expandtwoargs}{%
94   \def\@expandtwoargs#1#2#3{%
95     \edef\reserved@a{\noexpand#1{#2}{#3}}%
96     \reserved@a
97   }%
98 }{}

```

2.4 Add

\AppendGraphicsExtensions

```

99 \newcommand*\AppendGraphicsExtensions{%
100   \@ifundefined{Gin@extensions}{%
101     \let\Gin@extensions\empty
102   }{}%
103   \@ifstar{\grfext@Append\grfext@Check}{\grfext@Append\grfext@@Add}%

```

```

104 }%

\grfext@Append
105 \def\grfext@Append#1#2{%
106   \let\grfext@Print\@gobble
107   \edef\grfext@next{%
108     \noexpand\grfext@Add\noexpand#1{%
109       \zap@space#2 \@empty
110     }\noexpand\Gin@extensions,}%}
111   }%
112   \grfext@next
113   \let\grfext@Print\grfext@@Print
114   \grfext@Print\AppendGraphicsExtensions
115 }

\PrependGraphicsExtensions
116 \newcommand*{\PrependGraphicsExtensions}{%
117   \@ifundefined{Gin@extensions}{%
118     \let\Gin@extensions\@empty
119   }{}%
120   \@ifstar{\grfext@Prepend\grfext@Check}{\grfext@Prepend\grfext@@Add}%
121 }%

\grfext@Prepend
122 \def\grfext@Prepend#1#2{%
123   \let\grfext@Print\@gobble
124   \edef\grfext@next{%
125     \noexpand\grfext@Add\noexpand#1{%
126       \zap@space#2 \@empty
127     }\noexpand\Gin@extensions}%
128   }%
129   \grfext@next
130   \let\grfext@Print\grfext@@Print
131   \grfext@Print\PrependGraphicsExtensions
132 }

\grfext@Add
133 \def\grfext@Add#1#2{%
134   #1#2}%
135 }

\grfext@@Add
136 \def\grfext@@Add#1#2#3{%
137   \RemoveGraphicsExtensions{#1}%
138   \ifx\Gin@extensions\@empty
139     \def\Gin@extensions{#1}%
140   \else
141     \edef\Gin@extensions{#2#1#3}%
142   \fi
143 }

```

2.5 Check

```

\grfext@Check
144 \def\grfext@Check#1{%
145   \let\grfext@tmp\@empty
146   \@for\grfext@ext:=#1\do{%
147     \@ifundefined{Gin@rule@\grfext@ext}{%
148       }{%
149         \ifx\grfext@tmp\@empty
150           \let\grfext@tmp\grfext@ext

```

```

151     \else
152     \edef\grfext@tmp{\grfext@tmp,\grfext@ext}%
153     \fi
154   }%
155 }%
156 \ifx\grfext@tmp\@empty
157   \def\grfext@next##1##2{%
158   \else
159     \edef\grfext@next{%
160       \noexpand\grfext@@Add{\grfext@tmp}%
161     }%
162   \fi
163   \grfext@next
164 }

```

2.6 Remove

\RemoveGraphicsExtensions

```

165 \newcommand*{\RemoveGraphicsExtensions}[1]{%
166   \@ifundefined{Gin@extensions}{%
167     \def\Gin@extensions{%
168     }{%
169       \edef\grfext@tmp{\zap@space#1 \@empty}%
170       \@for\grfext@ext:=\grfext@tmp\do{%
171         \def\grfext@next{%
172           \let\grfext@tmp\Gin@extensions
173           \@expandtwoargs
174           \@removeelement\grfext@ext\Gin@extensions\Gin@extensions
175           \ifx\grfext@tmp\Gin@extensions
176             \let\grfext@next\relax
177           \fi
178           \grfext@next
179         }%
180         \grfext@next
181       }%
182     }%
183     \grfext@Print\RemoveGraphicsExtensions
184 }

```

2.7 Print

```

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

```

\PrintGraphicsExtensions

```

186 \def\PrintGraphicsExtensions{%
187   \grfext@Print\PrintGraphicsExtensions
188 }

```

\grfext@Print

```

189 \def\grfext@Print#1{%
190   \@PackageInfo{grfext}{%
191     Graphics extension search list:\MessageBreak
192     \@ifundefined{Gin@extensions}{%
193       <unavailable>%
194     }{%
195       [\Gin@extensions]%
196     }\MessageBreak
197     \string#1%
198   }%
199 }

```

`\grfext@@Print`

```
200 \let\grfext@@Print\grfext@Print
201 \grfext@AtEnd
202 \end{package}
```

3 Test

Load miniltx for standard catcode test.

```
203 \test1
204 \input miniltx.tex\relax
205 \resetatcatcode
206 \test1
```

3.1 Catcode checks for loading

```
207 \test1
208 \catcode'\{=1 %
209 \catcode'\}=2 %
210 \catcode'\#=6 %
211 \catcode'\@=11 %
212 \expandafter\ifx\csname count@\endcsname\relax
213   \countdef\count@=255 %
214 \fi
215 \expandafter\ifx\csname @gobble\endcsname\relax
216   \long\def\@gobble#1{ }%
217 \fi
218 \expandafter\ifx\csname @firstofone\endcsname\relax
219   \long\def\@firstofone#1{#1}%
220 \fi
221 \expandafter\ifx\csname loop\endcsname\relax
222   \expandafter\@firstofone
223 \else
224   \expandafter\@gobble
225 \fi
226 {%
227   \def\loop#1\repeat{%
228     \def\body{#1}%
229     \iterate
230   }%
231   \def\iterate{%
232     \body
233     \let\next\iterate
234   \else
235     \let\next\relax
236   \fi
237   \next
238 }%
239 \let\repeat=\fi
240}%
241 \def\RestoreCatcodes{}
242 \count@=0 %
243 \loop
244   \edef\RestoreCatcodes{%
245     \RestoreCatcodes
246     \catcode\the\count@=\the\catcode\count@\relax
247   }%
248 \ifnum\count@<255 %
249   \advance\count@ 1 %
250 \repeat
251
```

```

252 \def\RangeCatcodeInvalid#1#2{%
253   \count@=#1\relax
254   \loop
255     \catcode\count@=15 %
256   \ifnum\count@<#2\relax
257     \advance\count@ 1 %
258   \repeat
259 }
260 \def\Test{%
261   \RangeCatcodeInvalid{0}{47}%
262   \RangeCatcodeInvalid{58}{64}%
263   \RangeCatcodeInvalid{91}{96}%
264   \RangeCatcodeInvalid{123}{255}%
265   \catcode'\@=12 %
266   \catcode'\=0 %
267   \catcode'\{=1 %
268   \catcode'\}=2 %
269   \catcode'\#=6 %
270   \catcode'\[=12 %
271   \catcode'\]=12 %
272   \catcode'\%=14 %
273   \catcode'\ =10 %
274   \catcode13=5 %
275   \input grfext.sty\relax
276   \RestoreCatcodes
277 }
278 \Test
279 \csname @@end\endcsname
280 \end
281 \</test1>

```

3.2 Macro tests

3.2.1 Preamble

```

282 \*test2\
283 \NeedsTeXFormat{LaTeX2e}
284 \nofiles
285 \documentclass{article}
286 \usepackage[pdftex]{graphics}
287 \usepackage{grfext}[2007/09/30]
288 \usepackage{qstest}
289 \IncludeTests{*}
290 \LogTests{log}{*}{*}
291 \makeatletter

```

3.2.2 Detection of unwanted space

```

292 \let\orig@qstest\qstest
293 \let\orig@endqstest\endqstest
294 \def\qstest#1#2{%
295   \orig@qstest{#1}{#2}%
296   \setbox0\hbox\bgroup\begingroup\ignorespaces
297 }
298 \def\endqstest{%
299   \endgroup\egroup
300   \Expect*{\the\wd0}{0.0pt}%
301   \orig@endqstest
302 }

```

3.2.3 Generic test macros

```

303 \def\Test{\ifnextchar\bgroup\TestInitList\TestInitCmd}%
304 \def\TestInitList#1{%
305   \def\Gin@extensions{#1}%

```



```

306 \@Test
307 }
308 \def\TestInitCmd#1{%
309   \let\Gin@extensions#1\relax
310   \@Test
311 }
312 \def\@Test#1#2{%
313   \TestCmd{#1}%
314   \expandafter\Expect\expandafter*{\Gin@extensions}{#2}%
315 }

```

3.2.4 Test for \AppendGraphicsExtensions

```

316 \begin{qstest}{\AppendGraphicsExtensions}{\AppendGraphicsExtensions}
317   \let\TestCmd\AppendGraphicsExtensions
318   \Test\@undefined{.test}{.test}%
319   \Test\relax{.test}{.test}%
320   \Test{}{.test}{.test}%
321   \Test{.foo}{.test}{.foo,.test}%
322   \Test{.foo,.bar}{.test}{.foo,.bar,.test}%
323   \Test{.test}{.test}{.test}%
324   \Test{.foo,.test}{.test}{.foo,.test}%
325   \Test{.test,.foo}{.test}{.foo,.test}%
326   \Test{.foo,.test,.bar}{.test}{.foo,.bar,.test}%
327   \Test{.foo}{.test,.bar}{.foo,.test,.bar}%
328   \Test{.foo}{.test,.foo}{.test,.foo}%
329   \Test{.foo}{.test,.bar,.x,.y}{.foo,.test,.bar,.x,.y}%
330   \def\TestCmd{\AppendGraphicsExtensions*}%
331   \Test\@undefined{.test}{}%
332   \Test\relax{.test}{}%
333   \Test{}{.test}{}%
334   \Test{.test}{.test}{.test}%
335   \Test{.foo}{.test}{.foo}%
336   \Test{.foo,.bar}{.test}{.foo,.bar}%
337   \Test{.test}{.test}{.test}%
338   \Test{.foo,.test}{.test}{.foo,.test}%
339   \Test{.test,.foo}{.test}{.test,.foo}%
340   \Test{.foo,.test,.bar}{.test}{.foo,.test,.bar}%
341   \Test{.foo}{.test,.bar}{.foo}%
342   \Test{.foo}{.test,.foo}{.foo}%
343   \DeclareGraphicsRule{.test}{test}{.test}{}%
344   \Test\@undefined{.test}{.test}%
345   \Test\relax{.test}{.test}%
346   \Test{}{.test}{.test}%
347   \Test{.foo}{.test}{.foo,.test}%
348   \Test{.foo,.bar}{.test}{.foo,.bar,.test}%
349   \Test{.test}{.test}{.test}%
350   \Test{.foo,.test}{.test}{.foo,.test}%
351   \Test{.test,.foo}{.test}{.foo,.test}%
352   \Test{.foo,.test,.bar}{.test}{.foo,.bar,.test}%
353   \Test{.foo}{.test,.bar}{.foo,.test}%
354   \Test{.foo}{.test,.foo}{.foo,.test}%
355   \DeclareGraphicsRule{.bar}{bar}{.bar}{}%
356   \Test{.foo}{.test,.bar}{.foo,.test,.bar}%
357   \Test{.bar}{.test,.bar}{.test,.bar}%
358   \Test{.bar}{.bar,.test}{.bar,.test}%
359 \end{qstest}

```

3.2.5 Test for \PrependGraphicsExtensions

```

360 \begin{qstest}{\PrependGraphicsExtensions}{\PrependGraphicsExtensions}
361   \let\TestCmd\PrependGraphicsExtensions
362   \Test\@undefined{.test}{.test}%
363   \Test\relax{.test}{.test}%
364   \Test{}{.test}{.test}%

```

```

365 \Test{.foo}{.test}{.test,.foo}%
366 \Test{.foo,.bar}{.test}{.test,.foo,.bar}%
367 \Test{.test}{.test}{.test}%
368 \Test{.foo,.test}{.test}{.test,.foo}%
369 \Test{.test,.foo}{.test}{.test,.foo}%
370 \Test{.foo,.test,.bar}{.test}{.test,.foo,.bar}%
371 \Test{.foo}{.test,.bar}{.test,.bar,.foo}%
372 \Test{.foo}{.test,.foo}{.test,.foo}%
373 \Test{.foo}{.test,.bar,.x,.y}{.test,.bar,.x,.y,.foo}%
374 \def\TestCmd{\PrependGraphicsExtensions*}%
375 \Test\@undefined{.test}{}%
376 \Test\relax{.test}{}%
377 \Test{}{.test}{}%
378 \Test{.test}{.test}{.test}%
379 \Test{.foo}{.test}{.foo}%
380 \Test{.foo,.bar}{.test}{.foo,.bar}%
381 \Test{.test}{.test}{.test}%
382 \Test{.foo,.test}{.test}{.foo,.test}%
383 \Test{.test,.foo}{.test}{.test,.foo}%
384 \Test{.foo,.test,.bar}{.test}{.foo,.test,.bar}%
385 \DeclareGraphicsRule{.test}{test}{.test}{}%
386 \Test\@undefined{.test}{.test}%
387 \Test\relax{.test}{.test}%
388 \Test{}{.test}{.test}%
389 \Test{.foo}{.test}{.test,.foo}%
390 \Test{.foo,.bar}{.test}{.test,.foo,.bar}%
391 \Test{.test}{.test}{.test}%
392 \Test{.foo,.test}{.test}{.test,.foo}%
393 \Test{.test,.foo}{.test}{.test,.foo}%
394 \Test{.foo,.test,.bar}{.test}{.test,.foo,.bar}%
395 \Test{.foo}{.test,.bar}{.test,.foo}%
396 \Test{.foo}{.test,.foo}{.test,.foo}%
397 \DeclareGraphicsRule{.bar}{bar}{.bar}{}%
398 \Test{.foo}{.test,.bar}{.test,.bar,.foo}%
399 \Test{.bar}{.test,.bar}{.test,.bar}%
400 \Test{.bar}{.bar,.test}{.bar,.test}%
401 \end{qstest}

```

3.2.6 Test for \RemoveGraphicsExtensions

```

402 \begin{qstest}{\RemoveGraphicsExtensions}{\RemoveGraphicsExtensions}
403 \let\TestCmd\RemoveGraphicsExtensions
404 \Test\@undefined{.test}{}%
405 \Test\relax{.test}{}%
406 \Test{}{.test}{}%
407 \Test{.foo}{.test}{.foo}%
408 \Test{.foo,.bar}{.test}{.foo,.bar}%
409 \Test{.test}{.test}{}%
410 \Test{.foo,.test}{.test}{.foo}%
411 \Test{.test,.foo}{.test}{.foo}%
412 \Test{.foo,.test,.bar}{.test}{.foo,.bar}%
413 \Test{.test,.test}{.test}{}%
414 \Test{.foo,.test,.bar,.test,.test,.bar}{.test}{.foo,.bar,.bar}%
415 \Test{.foo,.test,.bar,.foo,.test,.bar}{.test,.bar}{.foo,.foo}%
416 \Test{.foo,.test,.bar,.foo,.test,.bar}{.test,.bar}{.foo,.foo}%
417 \end{qstest}

418 \begin{document}
419 \end{document}
420 </test2>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[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:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX 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-T_EX:

```
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):

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

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 T_EX distribution (teT_EX, miK_TE_X, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

¹<http://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 `pdf \LaTeX` :

```
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		118, 126, 138, 145, 149, 156, 169
<code>\#</code>	210, 269	<code>\@expandtwoargs</code> 93, 173
<code>\%</code>	272	<code>\@firstofone</code> 219, 222
<code>\@</code>	211, 265	<code>\@for</code> 146, 170
<code>\@PackageInfo</code>	190	<code>\@gobble</code> 106, 123, 216, 224
<code>\@Test</code>	306, 310, 312	<code>\@ifnextchar</code> 303
<code>\@empty</code>	101, 109,	<code>\@ifstar</code> 103, 120

326, 327, 328, 329, 331, 332,	\the 66, 67, 68, 69, 80, 246, 300
333, 334, 335, 336, 337, 338,	\TMP@EnsureCode 77,
339, 340, 341, 342, 344, 345,	84, 85, 86, 87, 88, 89, 90, 91, 92
346, 347, 348, 349, 350, 351,	
352, 353, 354, 356, 357, 358,	U
362, 363, 364, 365, 366, 367,	\usepackage 286, 287, 288
368, 369, 370, 371, 372, 373,	
375, 376, 377, 378, 379, 380,	W
381, 382, 383, 384, 386, 387,	\wd 300
388, 389, 390, 391, 392, 393,	\write 23, 47
394, 395, 396, 398, 399, 400,	
404, 405, 406, 407, 408, 409,	X
410, 411, 412, 413, 414, 415, 416	\x 8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72
\TestCmd . . 313, 317, 330, 361, 374, 403	
\TestInitCmd 303, 308	Z
\TestInitList 303, 304	\zap@space 109, 126, 169