

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	5
2.7	Print	6
3	Test	6
3.1	Catcode checks for loading	6
3.2	Macro tests	7
3.2.1	Preamble	7
3.2.2	Detection of unwanted space	7
3.2.3	Generic test macros	8
3.2.4	Test for <code>\AppendGraphicsExtensions</code>	8
3.2.5	Test for <code>\PrependGraphicsExtensions</code>	9
3.2.6	Test for <code>\RemoveGraphicsExtensions</code>	9
4	Installation	10
4.1	Download	10
4.2	Bundle installation	10
4.3	Package installation	10
4.4	Refresh file name databases	11
4.5	Some details for the interested	11
5	References	11
6	History	11
	[2007/09/30 v1.0]	11
7	Index	11

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 Reload check and identification

Reload 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 \expandafter\ifx\csname PackageInfo\endcsname\relax
19 \def\x#1#2{%
20 \immediate\write-1{Package #1 Info: #2.}%
21 }%
22 \else
23 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24 \fi
25 \x{grfext}{The package is already loaded}%
26 \endgroup
27 \expandafter\endinput
28 \fi
29 \endgroup

```

Package identification:

```

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

```

```

53 \ProvidesPackage{grfext}%
54 [2007/09/30 v1.0 Managing graphics extensions (H0)]

```

2.2 Catcodes

```

55 \expandafter\edef\csname grfext@AtEnd\endcsname{%
56   \catcode64 \the\catcode64\relax
57 }
58 \catcode64 11 % @
59 \def\TMP@EnsureCode#1#2{%
60   \edef\grfext@AtEnd{%
61     \grfext@AtEnd
62     \catcode#1 \the\catcode#1\relax
63   }%
64   \catcode#1 #2\relax
65 }
66 \TMP@EnsureCode{42}{12}% *
67 \TMP@EnsureCode{44}{12}% ,
68 \TMP@EnsureCode{47}{12}% /
69 \TMP@EnsureCode{58}{12}% :
70 \TMP@EnsureCode{60}{12}% <
71 \TMP@EnsureCode{61}{12}% =
72 \TMP@EnsureCode{62}{12}% >
73 \TMP@EnsureCode{91}{12}% [
74 \TMP@EnsureCode{93}{12}% ]

```

2.3 plain-TeX

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

```

75 \@ifundefined{@expandtwoargs}{%
76   \def\@expandtwoargs#1#2#3{%
77     \edef\reserved@a{\noexpand#1{#2}{#3}}%
78     \reserved@a
79   }%
80 }{}

```

2.4 Add

\AppendGraphicsExtensions

```

81 \newcommand*{\AppendGraphicsExtensions}{%
82   \@ifundefined{Gin@extensions}{%
83     \let\Gin@extensions\empty
84   }{}%
85   \@ifstar{\grfext@Append\grfext@Check}{\grfext@Append\grfext@@Add}%
86 }%

```

\grfext@Append

```

87 \def\grfext@Append#1#2{%
88   \let\grfext@Print\@gobble
89   \edef\grfext@next{%
90     \noexpand\grfext@Add\noexpand#1{%
91       \zap@space#2 \@empty
92     }\noexpand\Gin@extensions,}%{}%
93   }%
94   \grfext@next
95   \let\grfext@Print\grfext@@Print
96   \grfext@Print\AppendGraphicsExtensions
97 }

```

\PrependGraphicsExtensions

```

98 \newcommand*{\PrependGraphicsExtensions}{%
99   \@ifundefined{Gin@extensions}{%

```

```

100     \let\Gin@extensions\@empty
101   }{}%
102   \@ifstar{\grfext@Prepend\grfext@Check}{\grfext@Prepend\grfext@@Add}%
103 }%

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

\grfext@Add
115 \def\grfext@Add#1#2{%
116   #1{#2}%
117 }

\grfext@@Add
118 \def\grfext@@Add#1#2#3{%
119   \RemoveGraphicsExtensions{#1}%
120   \ifx\Gin@extensions\@empty
121     \def\Gin@extensions{#1}%
122   \else
123     \edef\Gin@extensions{#2#1#3}%
124   \fi
125 }

```

2.5 Check

```

\grfext@Check
126 \def\grfext@Check#1{%
127   \let\grfext@tmp\@empty
128   \@for\grfext@ext:=#1\do{%
129     \ifundefined{Gin@rule@\grfext@ext}{%
130       }{%
131         \ifx\grfext@tmp\@empty
132           \let\grfext@tmp\grfext@ext
133         \else
134           \edef\grfext@tmp{\grfext@tmp,\grfext@ext}%
135         \fi
136       }%
137     }%
138     \ifx\grfext@tmp\@empty
139       \def\grfext@next##1##2{%
140     \else
141       \edef\grfext@next{%
142         \noexpand\grfext@@Add{\grfext@tmp}%
143       }%
144     \fi
145     \grfext@next
146 }

```

2.6 Remove

\RemoveGraphicsExtensions

```

147 \newcommand*{\RemoveGraphicsExtensions}[1]{%
148   \@ifundefined{Gin@extensions}{%
149     \def\Gin@extensions{%
150   }{%
151     \edef\grfext@tmp{\zap@space#1 \@empty}%
152     \@for\grfext@ext:=\grfext@tmp\do{%
153       \def\grfext@next{%
154         \let\grfext@tmp\Gin@extensions
155         \@expandtwoargs
156         \@removeelement\grfext@ext\Gin@extensions\Gin@extensions
157         \ifx\grfext@tmp\Gin@extensions
158           \let\grfext@next\relax
159         \fi
160       }\grfext@next
161     }%
162     \grfext@next
163   }%
164 }%
165 \grfext@Print\RemoveGraphicsExtensions
166 }

```

2.7 Print

```

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

```

\PrintGraphicsExtensions

```

168 \def\PrintGraphicsExtensions{%
169   \grfext@Print\PrintGraphicsExtensions
170 }

```

\grfext@Print

```

171 \def\grfext@Print#1{%
172   \@PackageInfo{grfext}{%
173     Graphics extension search list:\MessageBreak
174     \@ifundefined{Gin@extensions}{%
175       <unavailable>%
176     }{%
177       [\Gin@extensions]%
178     }\MessageBreak
179     \string#1%
180   }%
181 }

```

\grfext@@Print

```

182 \let\grfext@@Print\grfext@Print

183 \grfext@AtEnd
184 \</package>

```

3 Test

Load miniltx for standard catcode test.

```

185 <*test1>
186 \input miniltx.tex\relax
187 \resetatcatcode
188 </test1>

```

3.1 Catcode checks for loading

```

189 <*test1>

```

```

190 \catcode'\@=11 %
191 \def\RestoreCatcodes{}
192 \count@=0 %
193 \loop
194   \edef\RestoreCatcodes{%
195     \RestoreCatcodes
196     \catcode\the\count@=\the\catcode\count@\relax
197   }%
198 \ifnum\count@<255 %
199   \advance\count@\@ne
200 \repeat
201
202 \def\RangeCatcodeInvalid#1#2{%
203   \count@=#1\relax
204   \loop
205     \catcode\count@=15 %
206     \ifnum\count@<#2\relax
207       \advance\count@\@ne
208     \repeat
209 }
210 \def\Test{%
211   \RangeCatcodeInvalid{0}{47}%
212   \RangeCatcodeInvalid{58}{64}%
213   \RangeCatcodeInvalid{91}{96}%
214   \RangeCatcodeInvalid{123}{255}%
215   \catcode'\@=12 %
216   \catcode'\=0 %
217   \catcode'\{=1 %
218   \catcode'\}=2 %
219   \catcode'\#=6 %
220   \catcode'\[=12 %
221   \catcode'\]=12 %
222   \catcode'\%=14 %
223   \catcode'\ =10 %
224   \catcode13=5 %
225   \input grfext.sty\relax
226   \RestoreCatcodes
227 }
228 \Test
229 \csname @@end\endcsname
230 \end
231 \test1

```

3.2 Macro tests

3.2.1 Preamble

```

232 \test2
233 \NeedsTeXFormat{LaTeX2e}
234 \nofiles
235 \documentclass{article}
236 \usepackage[pdftex]{graphics}
237 \usepackage{grfext}[2007/09/30]
238 \usepackage{qstest}
239 \IncludeTests{*}
240 \LogTests{log}{*}{*}
241 \makeatletter

```

3.2.2 Detection of unwanted space

```

242 \let\orig@qstest\qstest
243 \let\orig@endqstest\endqstest
244 \def\qstest#1#2{%
245   \orig@qstest{#1}{#2}%

```

```

246 \setbox0\hbox\bgroup\beginingroup\ignorespaces
247 }
248 \def\endqstest{%
249 \endgroup\egroup
250 \Expect*{\the\wd0}{0.0pt}%
251 \orig@endqstest
252 }

```

3.2.3 Generic test macros

```

253 \def\Test{\@ifnextchar\bgroup\TestInitList\TestInitCmd}%
254 \def\TestInitList#1{%
255 \def\Gin@extensions{#1}%
256 \@Test
257 }
258 \def\TestInitCmd#1{%
259 \let\Gin@extensions#1\relax
260 \@Test
261 }
262 \def\@Test#1#2{%
263 \TestCmd{#1}%
264 \expandafter\Expect\expandafter*\Gin@extensions{#2}%
265 }

```

3.2.4 Test for \AppendGraphicsExtensions

```

266 \begin{qstest}\AppendGraphicsExtensions\AppendGraphicsExtensions}
267 \let\TestCmd\AppendGraphicsExtensions
268 \Test\@undefined{.test}{.test}%
269 \Test\relax{.test}{.test}%
270 \Test{}{.test}{.test}%
271 \Test{.foo}{.test}{.foo,.test}%
272 \Test{.foo,.bar}{.test}{.foo,.bar,.test}%
273 \Test{.test}{.test}{.test}%
274 \Test{.foo,.test}{.test}{.foo,.test}%
275 \Test{.test,.foo}{.test}{.foo,.test}%
276 \Test{.foo,.test,.bar}{.test}{.foo,.bar,.test}%
277 \Test{.foo}{.test,.bar}{.foo,.test,.bar}%
278 \Test{.foo}{.test,.foo}{.test,.foo}%
279 \Test{.foo}{.test,.bar,.x,.y}{.foo,.test,.bar,.x,.y}%
280 \def\TestCmd{\AppendGraphicsExtensions*}%
281 \Test\@undefined{.test}{}%
282 \Test\relax{.test}{}%
283 \Test{}{.test}{}%
284 \Test{.test}{.test}{.test}%
285 \Test{.foo}{.test}{.foo}%
286 \Test{.foo,.bar}{.test}{.foo,.bar}%
287 \Test{.test}{.test}{.test}%
288 \Test{.foo,.test}{.test}{.foo,.test}%
289 \Test{.test,.foo}{.test}{.test,.foo}%
290 \Test{.foo,.test,.bar}{.test}{.foo,.test,.bar}%
291 \Test{.foo}{.test,.bar}{.foo}%
292 \Test{.foo}{.test,.foo}{.foo}%
293 \DeclareGraphicsRule{.test}{test}{.test}{}%
294 \Test\@undefined{.test}{.test}%
295 \Test\relax{.test}{.test}%
296 \Test{}{.test}{.test}%
297 \Test{.foo}{.test}{.foo,.test}%
298 \Test{.foo,.bar}{.test}{.foo,.bar,.test}%
299 \Test{.test}{.test}{.test}%
300 \Test{.foo,.test}{.test}{.foo,.test}%
301 \Test{.test,.foo}{.test}{.foo,.test}%
302 \Test{.foo,.test,.bar}{.test}{.foo,.bar,.test}%
303 \Test{.foo}{.test,.bar}{.foo,.test}%
304 \Test{.foo}{.test,.foo}{.foo,.test}%

```



```

305 \DeclareGraphicsRule{.bar}{bar}{.bar}{}%
306 \Test{.foo}{.test,.bar}{.foo,.test,.bar}%
307 \Test{.bar}{.test,.bar}{.test,.bar}%
308 \Test{.bar}{.bar,.test}{.bar,.test}%
309 \end{qstest}

```

3.2.5 Test for \PrependGraphicsExtensions

```

310 \begin{qstest}{PrependGraphicsExtensions}{PrependGraphicsExtensions}
311 \let\TestCmd\PrependGraphicsExtensions
312 \Test\@undefined{.test}{.test}%
313 \Test\relax{.test}{.test}%
314 \Test{}{.test}{.test}%
315 \Test{.foo}{.test}{.test,.foo}%
316 \Test{.foo,.bar}{.test}{.test,.foo,.bar}%
317 \Test{.test}{.test}{.test}%
318 \Test{.foo,.test}{.test}{.test,.foo}%
319 \Test{.test,.foo}{.test}{.test,.foo}%
320 \Test{.foo,.test,.bar}{.test}{.test,.foo,.bar}%
321 \Test{.foo}{.test,.bar}{.test,.bar,.foo}%
322 \Test{.foo}{.test,.foo}{.test,.foo}%
323 \Test{.foo}{.test,.bar,.x,.y}{.test,.bar,.x,.y,.foo}%
324 \def\TestCmd{\PrependGraphicsExtensions*}%
325 \Test\@undefined{.test}{}%
326 \Test\relax{.test}{}%
327 \Test{}{.test}{}%
328 \Test{.test}{.test}{.test}%
329 \Test{.foo}{.test}{.foo}%
330 \Test{.foo,.bar}{.test}{.foo,.bar}%
331 \Test{.test}{.test}{.test}%
332 \Test{.foo,.test}{.test}{.foo,.test}%
333 \Test{.test,.foo}{.test}{.test,.foo}%
334 \Test{.foo,.test,.bar}{.test}{.foo,.test,.bar}%
335 \DeclareGraphicsRule{.test}{test}{.test}{}%
336 \Test\@undefined{.test}{.test}%
337 \Test\relax{.test}{.test}%
338 \Test{}{.test}{.test}%
339 \Test{.foo}{.test}{.test,.foo}%
340 \Test{.foo,.bar}{.test}{.test,.foo,.bar}%
341 \Test{.test}{.test}{.test}%
342 \Test{.foo,.test}{.test}{.test,.foo}%
343 \Test{.test,.foo}{.test}{.test,.foo}%
344 \Test{.foo,.test,.bar}{.test}{.test,.foo,.bar}%
345 \Test{.foo}{.test,.bar}{.test,.foo}%
346 \Test{.foo}{.test,.foo}{.test,.foo}%
347 \DeclareGraphicsRule{.bar}{bar}{.bar}{}%
348 \Test{.foo}{.test,.bar}{.test,.bar,.foo}%
349 \Test{.bar}{.test,.bar}{.test,.bar}%
350 \Test{.bar}{.bar,.test}{.bar,.test}%
351 \end{qstest}

```

3.2.6 Test for \RemoveGraphicsExtensions

```

352 \begin{qstest}{RemoveGraphicsExtensions}{RemoveGraphicsExtensions}
353 \let\TestCmd\RemoveGraphicsExtensions
354 \Test\@undefined{.test}{}%
355 \Test\relax{.test}{}%
356 \Test{}{.test}{}%
357 \Test{.foo}{.test}{.foo}%
358 \Test{.foo,.bar}{.test}{.foo,.bar}%
359 \Test{.test}{.test}{}%
360 \Test{.foo,.test}{.test}{.foo}%
361 \Test{.test,.foo}{.test}{.foo}%
362 \Test{.foo,.test,.bar}{.test}{.foo,.bar}%
363 \Test{.test,.test}{.test}{}%

```

```

364 \Test{.foo,.test,.bar,.test,.test,.bar}{.test}{.foo,.bar,.bar}%
365 \Test{.foo,.test,.bar,.foo,.test,.bar}{.test,.bar}{.foo,.foo}%
366 \Test{.foo,.test,.bar,.foo,.test,.bar}{.test,.bar}{.foo,.foo}%
367 \end{qstest}

368 \begin{document}
369 \end{document}
370 </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 \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.

¹<http://ftp.ctan.org/tex-archive/>

4.4 Refresh file name databases

If your \TeX distribution (te \TeX , mik \TeX , ...) relies on file name databases, you must refresh these. For example, te \TeX users run `texhash` or `mktextlsr`.

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	\%	222
\#	219	\@ 190, 215

276, 277, 278, 279, 281, 282,	\the 56, 62, 196, 250
283, 284, 285, 286, 287, 288,	\TMP@EnsureCode 59,
289, 290, 291, 292, 294, 295,	66, 67, 68, 69, 70, 71, 72, 73, 74
296, 297, 298, 299, 300, 301,	
302, 303, 304, 306, 307, 308,	U
312, 313, 314, 315, 316, 317,	\usepackage 236, 237, 238
318, 319, 320, 321, 322, 323,	
325, 326, 327, 328, 329, 330,	W
331, 332, 333, 334, 336, 337,	\wd 250
338, 339, 340, 341, 342, 343,	\write 20, 41
344, 345, 346, 348, 349, 350,	
354, 355, 356, 357, 358, 359,	X
360, 361, 362, 363, 364, 365, 366	\x 8, 10, 12, 19, 23, 25, 40, 45, 52
\TestCmd . . 263, 267, 280, 311, 324, 353	
\TestInitCmd 253, 258	Z
\TestInitList 253, 254	\zap@space 91, 108, 151