

The hopatch package

Heiko Oberdiek

<heiko.oberdiek at gmail.com>

2011/06/24 v1.1

Abstract

This packages provides a wrapper to various package hooks provided by other packages or classes, but does not define own hooks.

Contents

1	Documentation	1
2	Implementation	3
2.1	Catcodes and package identification	3
2.2	Resources	3
2.3	Package patching	4
3	Test	5
3.1	Catcode checks for loading	5
4	Installation	8
4.1	Download	8
4.2	Bundle installation	8
4.3	Package installation	8
4.4	Refresh file name databases	9
4.5	Some details for the interested	9
5	Catalogue	9
6	References	10
7	History	10
	[2011/01/30 v1.0]	10
	[2011/06/24 v1.1]	10
8	Index	11

1 Documentation

Sometimes I want to add code right after a package has been loaded. Examples are bug fixes, adaptations, or added features as needed by package `hyperref`, for instance.

Unhappily \LaTeX does not provide this kind of hook. `\AtEndOfPackage` can be used inside the package only, because \LaTeX clears the hook right before it loads the package.

However, there are already many packages and classes that provide hooks that are executed after the package is loaded, see table 1.

Table 1: After package hooking

Macro	Provider
<code>\AfterPackage</code>	package <code>scrfile</code> [5]
<code>\AtEndOfPackageFile</code>	package <code>filehook</code> [2]
<code>\AtEndPackage</code>	class <code>memoir</code> [4]

Table 2: After begin document hooking

Macro	Provider
<code>\AtBeginDocument</code>	L ^A T _E X's kernel
<code>\AtEndPreamble</code>	package <code>etoolbox</code> [1]
<code>\AfterEndPreamble</code>	package <code>etoolbox</code>

Package `hopatch` can be used without the packages of table 1. But for an early executing right after a package is loaded, one of the following class or packages should be loaded before using `\hopatch@AfterPackage`:

- package `filehook`
- package `scrfile`
- class `memoir`

Therefore I skip writing a new package for hooking into L^AT_EX's package management and use this package to provide a wrapper to patch a package after it is loaded.

`\hopatch@AfterPackage {⟨package⟩} {⟨patch code⟩}`

If the package is already loaded, the *⟨patch code⟩* is executed immediately. Otherwise the *⟨patch code⟩* is stored in a command and tried at later locations until the package is available.

The patch is tried in the following order:

1. If the package is already loaded, the patch is applied immediately. Further locations are not tried.
2. `\AtEndPackage`, provided by class `memoir` [4], and `\AfterPackage`, provided by package `scrfile` [5], are called right after the package file is input before the hook of L^AT_EX's `\AtEndOfPackage`.
3. `\AtEndOfPackageFile`, provided by package `filehook` [2], is called after the package is loaded and after the hook of L^AT_EX's `\AtEndOfPackage`.
4. `\AtEndPreamble`, provided by package `etoolbox` [1], is called at the beginning of `\begin{document}` before the hook of L^AT_EX's `\AtBeginDocument`.
5. `\AtBeginDocument`, provided by L^AT_EX.
6. `\AfterEndDocument`, provided by package `etoolbox` [1], is called at the very end of `\begin{document}`. Preamble commands are already forbidden there.

Because of the various locations the patch code is restricted to limitations:

- Preamble commands, see L^AT_EX's `\onlypreamble` throw an error if used after `\begin{document}`. This is already the case for `\AfterEndDocument`. Therefore preamble commands are forbidden in the patching code. There are four exceptions `\ifpackageloaded`, `\ifclassloaded`, `\ifpackagelater` and `\ifclasslater`. They are redefined during `\AfterEndDocument` using the counterparts of package `ltxcmds` [3].

- `\AfterPackage` of package `scrfile` and `\AtEndPackage` of class `memoir` call the hook before L^AT_EX's `\AtEndOfPackage`.

2 Implementation

```
1 \*package\
```

2.1 Catcodes and package identification

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^~M
4 \endlinechar=13 %
5 \catcode123=1 % {
6 \catcode125=2 % }
7 \catcode64=11 % @
8 \def\x{\endgroup
9 \expandafter\edef\csname H0patch@AtEnd\endcsname{%
10 \endlinechar=\the\endlinechar\relax
11 \catcode13=\the\catcode13\relax
12 \catcode32=\the\catcode32\relax
13 \catcode35=\the\catcode35\relax
14 \catcode61=\the\catcode61\relax
15 \catcode64=\the\catcode64\relax
16 \catcode123=\the\catcode123\relax
17 \catcode125=\the\catcode125\relax
18 }%
19 }%
20 \x\catcode61\catcode48\catcode32=10\relax%
21 \catcode13=5 % ^~M
22 \endlinechar=13 %
23 \catcode35=6 % #
24 \catcode64=11 % @
25 \catcode123=1 % {
26 \catcode125=2 % }
27 \def\TMP@EnsureCode#1#2{%
28 \edef\H0patch@AtEnd{%
29 \H0patch@AtEnd
30 \catcode#1=\the\catcode#1\relax
31 }%
32 \catcode#1=#2\relax
33 }
34 \TMP@EnsureCode{40}{12}% (
35 \TMP@EnsureCode{41}{12}% )
36 \TMP@EnsureCode{43}{12}% +
37 \TMP@EnsureCode{46}{12}% .
38 \TMP@EnsureCode{47}{12}% /
39 \TMP@EnsureCode{91}{12}% [
40 \TMP@EnsureCode{93}{12}% ]
41 \edef\H0patch@AtEnd{\H0patch@AtEnd\noexpand\endinput}

Package identification.
42 \NeedsTeXFormat{LaTeX2e}
43 \ProvidesPackage{hopatch}%
44 [2011/06/24 v1.1 Wrapper for package hooks (H0)]
```

2.2 Resources

```
45 \begingroup\expandafter\expandafter\expandafter\endgroup
46 \expandafter\ifx\csname RequirePackage\endcsname\relax
47 \def\TMP@RequirePackage#1[#2]{%
48 \begingroup\expandafter\expandafter\expandafter\endgroup
49 \expandafter\ifx\csname ver@#1.sty\endcsname\relax
50 \input #1.sty\relax
51 \fi
```

```

52 }%
53 \TMP@RequirePackage{ltxcmds}[2010/12/12]%
54 \else
55 \RequirePackage{ltxcmds}[2010/12/12]%
56 \fi

\H0patch@counter

57 \def\H0patch@counter{0}%

\H0patch@StepCounter

58 \ltx@ifundefined{numexpr}{%
59 \def\H0patch@StepCounter{%
60 \begingroup
61 \count@\H0patch@counter\relax
62 \advance\count@\ltx@one\relax
63 \edef\x{\endgroup
64 \noexpand\def\noexpand\H0patch@counter{\the\count@}%
65 }%
66 }%
67 }{%
68 \def\H0patch@StepCounter{%
69 \edef\H0patch@counter{%
70 \the\numexpr\H0patch@counter+\ltx@one\relax
71 }%
72 }%
73 }

\H0patch@list

74 \def\H0patch@list{}

\H0patch@Add

75 \def\H0patch@Add{%
76 \ltx@LocalAppendToMacro\H0patch@list
77 }

```

2.3 Package patching

```

\hopatch@AfterPackage

78 \def\hopatch@AfterPackage#1{%
79 \ltx@ifpackageloaded{#1}{%
80 \ltx@firstofone
81 }{%
82 \H0patch@AfterPackage{#1}%
83 }%
84 }

\H0patch@AfterPackage

85 \def\H0patch@AfterPackage#1{%
86 \edef\H0patch@temp{#1}%
87 \H0patch@StepCounter
88 \expandafter\H0patch@@@AfterPackage
89 \csname H0patch@\H0patch@counter\expandafter\endcsname{%
90 \H0patch@temp
91 }%
92 }

\H0patch@@@AfterPackage

93 \def\H0patch@@@AfterPackage#1#2#3{%
94 \begingroup
95 \toks@{#3}%
96 \xdef\H0patch@gtemp{%
97 \noexpand\ltx@ifpackageloaded{#2}{%

```

```

98      \noexpand\let\noexpand#1\noexpand\relax
99      \the\toks@
100    }{}%
101  }%
102 \endgroup
103 \let#1\H0patch@gtemp
104 \H0patch@Add#1%
105 \H0patch@Try{AfterPackage}{#2}#1%
106 \H0patch@Try{AtEndPackage}{#2}#1%
107 \H0patch@Try{AtEndOfPackageFile}{#2}#1%
108 }

\H0patch@Try
109 \def\H0patch@Try#1#2#3{%
110   \ltx@ifundefined{#1}{}{%
111     \csname #1\endcsname{#2}{#3}%
112   }%
113 }

114 \AtBeginDocument{\H0patch@list}
115 \ltx@ifundefined{AtEndPreamble}{}{%
116   \ltx@ifundefined{@endpreamblehook}{}{%
117     \AtEndPreamble{\H0patch@list}%
118   }%
119 }
120 \ltx@ifundefined{AfterEndPreamble}{}{%
121   \ltx@ifundefined{@afterendpreamblehook}{}{%
122     \AfterEndPreamble{%
123       \let\H0patch@OrgIfPackageLoaded\ifpackageloaded
124       \let\H0patch@OrgIfPackageLater\ifpackagelater
125       \let\H0patch@OrgIfClassLoaded\ifclassloaded
126       \let\H0patch@OrgIfClassLater\ifclasslater
127       \let\@ifpackageloaded\ltx@ifpackageloaded
128       \let\@ifpackagelater\ltx@ifpackagelater
129       \let\@ifclassloaded\ltx@ifclassloaded
130       \let\@ifclasslater\ltx@ifclasslater
131       \H0patch@list
132       \let\@ifpackageloaded\H0patch@OrgIfPackageLoaded
133       \let\@ifpackagelater\H0patch@OrgIfPackageLater
134       \let\@ifclassloaded\H0patch@OrgIfClassLoaded
135       \let\@ifclasslater\H0patch@OrgIfClassLater
136     }%
137   }%
138 }

139 \H0patch@AtEnd%
140 </package>

```

3 Test

```

141 <*test1>
142 \def\LoadCommand{\RequirePackage{hopatch}[2011/06/24]}
143 </test1>

```

3.1 Catcode checks for loading

```

144 <*test1>
145 \catcode`\{=1 %
146 \catcode`\}=2 %
147 \catcode`\#=6 %
148 \catcode`\@=11 %
149 \expandafter\ifx\csname count@\endcsname\relax

```

```

150 \countdef\count@=255 %
151 \fi
152 \expandafter\ifx\csname @gobble\endcsname\relax
153 \long\def\@gobble#1{%
154 \fi
155 \expandafter\ifx\csname @firstofone\endcsname\relax
156 \long\def\@firstofone#1{#1}%
157 \fi
158 \expandafter\ifx\csname loop\endcsname\relax
159 \expandafter\@firstofone
160 \else
161 \expandafter\@gobble
162 \fi
163 {%
164 \def\loop#1\repeat{%
165 \def\body{#1}%
166 \iterate
167 }%
168 \def\iterate{%
169 \body
170 \let\next\iterate
171 \else
172 \let\next\relax
173 \fi
174 \next
175 }%
176 \let\repeat=\fi
177 }%
178 \def\RestoreCatcodes{}
179 \count@=0 %
180 \loop
181 \edef\RestoreCatcodes{%
182 \RestoreCatcodes
183 \catcode\the\count@=\the\catcode\count@\relax
184 }%
185 \ifnum\count@<255 %
186 \advance\count@ 1 %
187 \repeat
188
189 \def\RangeCatcodeInvalid#1#2{%
190 \count@=#1\relax
191 \loop
192 \catcode\count@=15 %
193 \ifnum\count@<#2\relax
194 \advance\count@ 1 %
195 \repeat
196 }
197 \def\RangeCatcodeCheck#1#2#3{%
198 \count@=#1\relax
199 \loop
200 \ifnum#3=\catcode\count@
201 \else
202 \errmessage{%
203 Character \the\count@\space
204 with wrong catcode \the\catcode\count@\space
205 instead of \number#3%
206 }%
207 \fi
208 \ifnum\count@<#2\relax
209 \advance\count@ 1 %
210 \repeat
211 }

```

```

212 \def\space{ }
213 \expandafter\ifx\csname LoadCommand\endcsname\relax
214 \def\LoadCommand{\input hopatch.sty\relax}%
215 \fi
216 \def\Test{%
217 \RangeCatcodeInvalid{0}{47}%
218 \RangeCatcodeInvalid{58}{64}%
219 \RangeCatcodeInvalid{91}{96}%
220 \RangeCatcodeInvalid{123}{255}%
221 \catcode`\@=12 %
222 \catcode`\=0 %
223 \catcode`\%=14 %
224 \LoadCommand
225 \RangeCatcodeCheck{0}{36}{15}%
226 \RangeCatcodeCheck{37}{37}{14}%
227 \RangeCatcodeCheck{38}{47}{15}%
228 \RangeCatcodeCheck{48}{57}{12}%
229 \RangeCatcodeCheck{58}{63}{15}%
230 \RangeCatcodeCheck{64}{64}{12}%
231 \RangeCatcodeCheck{65}{90}{11}%
232 \RangeCatcodeCheck{91}{91}{15}%
233 \RangeCatcodeCheck{92}{92}{0}%
234 \RangeCatcodeCheck{93}{96}{15}%
235 \RangeCatcodeCheck{97}{122}{11}%
236 \RangeCatcodeCheck{123}{255}{15}%
237 \RestoreCatcodes
238 }
239 \Test
240 \csname @@end\endcsname
241 \end
242 </test1>
243 (*test2)
244 \NeedsTeXFormat{LaTeX2e}
245 \providecommand\variant{0}
246 \RequirePackage{filecontents}
247 \begin{filecontents}{foo.sty}
248 \ProvidesPackage{foo}
249 \def\msg#{\immediate\write16}
250 \def\foo#1{%
251 \msg{\fooformat{#1}}%
252 }
253 \def\fooformat#1{[#1]}% hash-ok
254 \foo{* Executing foo at package loading}
255 \end{filecontents}
256
257 \ifnum\variant=1 %
258 \documentclass{memoir}%
259 \else
260 \documentclass{article}%
261 \fi
262
263 \ifcase\variant\relax
264 \or % 1
265 \or % 2
266 \usepackage{etoolbox}%
267 \or % 3
268 \usepackage{scrfile}%
269 \or % 4
270 \usepackage{filehook}%
271 \fi
272
273 \AtBeginDocument{\foo{* AtBeginDocument before hopatch}}

```

```

274 \usepackage{hopatch}
275 \AtBeginDocument{\foo{* AtBeginDocument after hopatch}}
276
277 \makeatletter
278 \hopatch@AfterPackage{foo}{%
279   \def\fooformat#1{<<#1>>}%
280 }
281 \makeatother
282
283 \AtBeginDocument{\foo{* AtBeginDocument before foo}}
284 \usepackage{foo}
285 \AtBeginDocument{\foo{* AtBeginDocument after foo}}
286
287 \foo{* Executing in preamble}
288
289 \begin{document}
290 \foo{* Executing in document}
291 \end{document}
292 </test2>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/hopatch.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/hopatch.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 hopatch.dtx
```

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

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

```
hopatch.sty          → tex/latex/oberdiek/hopatch.sty
hopatch.pdf          → doc/latex/oberdiek/hopatch.pdf
test/hopatch-test1.tex → doc/latex/oberdiek/test/hopatch-test1.tex
test/hopatch-test2.tex → doc/latex/oberdiek/test/hopatch-test2.tex
hopatch.dtx          → source/latex/oberdiek/hopatch.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 (`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 hopatch.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{hopatch.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 hopatch.dtx
makeindex -s gind.ist hopatch.idx
pdflatex hopatch.dtx
makeindex -s gind.ist hopatch.idx
pdflatex hopatch.dtx
```

5 Catalogue

The following XML file can be used as source for the [\$\TeX\$ Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `hopatch.xml`.

293 `<*`catalogue>

```

294 <?xml version='1.0' encoding='us-ascii'?>
295 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
296 <entry datestamp='$Date$' modifier='$Author$' id='hopatch'>
297   <name>hopatch</name>
298   <caption>Load patches for packages.</caption>
299   <authorref id='auth:oberdiek'></authorref>
300   <copyright owner='Heiko Oberdiek' year='2011'></copyright>
301   <license type='lppl1.3'></license>
302   <version number='1.1'></version number>
303   <description>
304     The hopatch package provides a command with which the user may
305     register a piece of patch code for a particular package. Hopatch
306     will apply the patch immediately, if the relevant package has
307     already been loaded; otherwise it will store the patch until the
308     package appears.
309   </p>
310   The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
311 </description>
312 <documentation details='Package documentation'
313   href='ctan:/macros/latex/contrib/oberdiek/hopatch.pdf'></documentation details>
314 <ctan file='true' path='/macros/latex/contrib/oberdiek/hopatch.dtx'></ctan file>
315 <miktex location='oberdiek'></miktex location>
316 <texlive location='oberdiek'></texlive location>
317 <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'></install path>
318 </entry>
319 </catalogue>

```

6 References

- [1] Philipp Lehman: *The etoolbox Package* 2011-01-03.
[CTAN:macros/latex/contrib/etoolbox/etoolbox.pdf](#)
- [2] Martin Scharrer: *The filehook Package*; 2011-01-09.
[CTAN:macros/latex/contrib/filehook/filehook.pdf](#)
- [3] Heiko Oberdiek: *The ltxcmds Package*; 2010-12-12.
[CTAN:macros/latex/contrib/oberdiek/ltxcmds.pdf](#)
- [4] Peter Wilson, Lars Madsen: *The Memoir Class for Configurable Typesetting, User Guide*; 2010. [CTAN:macros/latex/contrib/memoir/memman.pdf](#)
- [5] Markus Kohm, Jens-Uwe Morawski: *The Guide KOMA-Script*; 2011-01-20.
[CTAN:macros/latex/contrib/koma-script/scrguien.pdf](#)

7 History

[2011/01/30 v1.0]

- First public version.

[2011/06/24 v1.1]

- Fix the use of `\AtEndPreamble` and `\AfterEndPreamble`. They are redefined by package `etoolbox` after their hooks are used and generate an error message then.

8 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.

Symbols	
\#	147
\%	223
\@	148, 221
\@firstofone	156, 159
\@gobble	153, 161
\@ifclasslater	126, 130, 135
\@ifclassloaded	125, 129, 134
\@ifpackagelater	124, 128, 133
\@ifpackageloaded	123, 127, 132
\\	222
\{	145
\}	146
A	
\advance	62, 186, 194, 209
\AfterEndPreamble	122
\AtBeginDocument	114, 273, 275, 283, 285
\AtEndPreamble	117
B	
\begin	247, 289
\body	165, 169
C	
\catcode	2, 3, 5, 6, 7, 11, 12, 13, 14, 15, 16, 17, 20, 21, 23, 24, 25, 26, 30, 32, 145, 146, 147, 148, 183, 192, 200, 204, 221, 222, 223
\count@	61, 62, 64, 150, 179, 183, 185, 186, 190, 192, 193, 194, 198, 200, 203, 204, 208, 209
\countdef	150
\csname	9, 46, 49, 89, 111, 149, 152, 155, 158, 213, 240
D	
\documentclass	258, 260
E	
\end	241, 255, 291
\endcsname	9, 46, 49, 89, 111, 149, 152, 155, 158, 213, 240
\endinput	41
\endlinechar	4, 10, 22
\errmessage	202
F	
\foo	250, 254, 273, 275, 283, 285, 287, 290
\fooformat	251, 253, 279
H	
\HOpatch@@AfterPackage	88, <u>93</u>
\HOpatch@Add	<u>75</u> , 104
\HOpatch@AfterPackage	82, <u>85</u>
\hopatch@AfterPackage	<u>2</u> , <u>78</u> , 278
\HOpatch@AtEnd	28, 29, 41, 139
\HOpatch@counter	<u>57</u> , 61, 64, 69, 70, 89
\HOpatch@gtemp	96, 103
\HOpatch@list	<u>74</u> , 76, 114, 117, 131
\HOpatch@OrgIfClassLater	126, 135
\HOpatch@OrgIfClassLoaded	125, 134
\HOpatch@OrgIfPackageLater	124, 133
\HOpatch@OrgIfPackageLoaded	123, 132
\HOpatch@StepCounter	<u>58</u> , 87
\HOpatch@temp	86, 90
\HOpatch@Try	105, 106, 107, <u>109</u>
I	
\ifcase	263
\ifnum	185, 193, 200, 208, 257
\ifx	46, 49, 149, 152, 155, 158, 213
\immediate	249
\input	50, 214
\iterate	166, 168, 170
L	
\LoadCommand	142, 214, 224
\loop	164, 180, 191, 199
\ltx@firstofone	80
\ltx@ifclasslater	130
\ltx@ifclassloaded	129
\ltx@ifpackagelater	128
\ltx@ifpackageloaded	79, 97, 127
\ltx@ifundefined	58, 110, 115, 116, 120, 121
\ltx@LocalAppendToMacro	76
\ltx@one	62, 70
M	
\makeatletter	277
\makeatother	281
\msg	249, 251
N	
\NeedsTeXFormat	42, 244
\next	170, 172, 174
\number	205
\numexpr	70
P	
\providecommand	245
\ProvidesPackage	43, 248
R	
\RangeCatcodeCheck	197, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236
\RangeCatcodeInvalid	189, 217, 218, 219, 220
\repeat	164, 176, 187, 195, 210
\RequirePackage	55, 142, 246
\RestoreCatcodes	178, 181, 182, 237
S	
\space	203, 204, 212

T		V	
\Test	216, 239	\variant	245, 257, 263
\the	10, 11, 12, 13, 14, 15, 16, 17, 30, 64, 70, 99, 183, 203, 204		
\TMP@EnsureCode		W	
.....	27, 34, 35, 36, 37, 38, 39, 40	\write	249
\TMP@RequirePackage	47, 53		
\toks@	95, 99		
U		X	
\usepackage ...	266, 268, 270, 274, 284	\x	8, 20, 63