

The rotchiffre package

Heiko Oberdiek*

2016/05/16 v1.1

Abstract

This package implements chiffres ROT13 with its variants ROT5, ROT18, and ROT47.

Contents

1	Documentation	2
1.1	Motivation	2
1.2	Usage	2
1.2.1	Examples	2
2	Implementation	3
2.1	Reload check and package identification	3
2.2	Catcodes	4
2.3	Loading resources	4
2.4	\EdefRot as robust macro	5
2.5	Set \lccode on a range of characters	5
2.6	Chiffres	6
2.6.1	ROT13	6
2.6.2	ROT5	6
2.6.3	ROT18	7
2.6.4	ROT47	7
2.7	\RotCh@rot with big char support	7
2.8	\RotCh@rot without big char support	8
3	Installation	8
3.1	Download	8
3.2	Bundle installation	9
3.3	Package installation	9
3.4	Refresh file name databases	9
3.5	Some details for the interested	9
4	References	10
5	History	10
	[2010/11/12 v1.0]	10
	[2016/05/16 v1.1]	10
6	Index	10

*Please report any issues at <https://github.com/ho-tex/oberdiek/issues>

1 Documentation

1.1 Motivation

In the newsgroup `comp.text.tex` there was a discussion [1] about package `fontspec`. Stephan Hennig provided an example to implement ROT13 as OpenType feature [2]. And Robin Fairbairns requested a CTAN upload [3] ☹.

But I think it would be not fair to the users of old \TeX engines without OpenType support that they will not be able to decrypt texts generated by the new package ☹. Therefore I have written this package that implements ROT13 even for `iniTeX`. Also other variants ROT5, ROT18, ROT47 are provided.

1.2 Usage

```
\EdefRot {<type>} {<cmd>} {<text>}
```

The `<text>` is expanded and sanitized. All tokens are letters with catcode 12 (other with the exception of the space token that has character code 32 (0x20) and catcode 10 (space)). This follows \TeX 's convention of `\string` and `\meaning`.

The chiffre type is specified by `<type>` it takes a number. For example, ROT13 is specified by 13. The selected chiffre is applied to `<text>` and the result is stored in macro `<cmd>`.

The following table lists the supported rotation chiffres.

chiffre	from	to
ROT13	A-Z	N-Z A-M
	a-z	n-z a-m
ROT5	0-9	5-9 0-4
ROT18	A-Z 0-9	S-Z 0-9 A-R
	a-z	n-z a-m
ROT47	!-~	P-~ !-0

In case of ROT47 the range is the ASCII range from character codes 33 (0x21) ‘!’ upto 126 (0xFE) ‘~’.

The specifications of the algorithms are taken from the description in Wikipedia [4, 5], ROT18 is further specified by “computerfreak” [6].

1.2.1 Examples

The famous English pangram [7] is converted by

```
\EdefRot{13}\result{The quick brown fox jumps over the lazy dog}
```

The result is stored in macro `\result` with the following contents:

```
Gur dhvpx oebja sbk whzcf bire gur ynml qbt
```

Command names are converted to strings before. Therefore the text should not contain \TeX markup, example:

```
\EdefRot{13}\result{\texttt{Hello}\par\textit{World}}
\result → Uryyb\nqinapr \cne@qrnguplpyrf \@ar Jbeyq
```

But macros can be used that contain text. They are expanded.

```
\newcommand{\Name}{Heiko}
\newcommand{\Email}{heiko.oberdiek at gmail.com}
\EdefRot{13}\result{Hello \Name\space<\Email>}
\result → Uryyb Urvxb <urvxb.boreqvrnx ng tbbtyrznvy.pbz>
```

2 Implementation

1 (*package)

2.1 Reload check and package identification

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

```
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@rotchiffre.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.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{rotchiffre}{The package is already loaded}%
29 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^~M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51 \def\x#1#2#3[#4]{\endgroup
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
```

```

57     #2[#{#3}]%
58     \ifx#1\@undefined
59         \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62         \xdef#1{#3}%
63     \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@rotchiffre.sty\endcsname
67 \ProvidesPackage{rotchiffre}%
68 [2016/05/16 v1.1 Perform simple rotation ciphers (H0)]%

```

2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^~M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname RotCh@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^~M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\RotCh@AtEnd{%
96     \RotCh@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{42}{12}% *
102 \TMP@EnsureCode{43}{12}% +
103 \TMP@EnsureCode{45}{12}% -
104 \TMP@EnsureCode{46}{12}% .
105 \TMP@EnsureCode{47}{12}% /
106 \TMP@EnsureCode{60}{12}% <
107 \TMP@EnsureCode{62}{12}% >
108 \TMP@EnsureCode{91}{12}% [
109 \TMP@EnsureCode{93}{12}% ]
110 \TMP@EnsureCode{96}{12}% ‘
111 \edef\RotCh@AtEnd{\RotCh@AtEnd\noexpand\endinput}

```

2.3 Loading resources

```

112 \begingroup\expandafter\expandafter\expandafter\endgroup
113 \expandafter\ifx\csname RequirePackage\endcsname\relax

```

```

114 \input infwarerr.sty\relax
115 \input ltxcmds.sty\relax
116 \input pdfescape.sty\relax
117 \else
118 \RequirePackage{infwarerr}[2010/04/08]%
119 \RequirePackage{ltxcmds}[2010/03/01]%
120 \RequirePackage{pdfescape}[2010/03/01]%
121 \fi

```

2.4 \EdefRot as robust macro

The main macro `\EdefRot` is made robust if ε -TeX or L^AT_EX are present.

```

\EdefRot
122 \ltx@ifundefined{protected}{%
123 \ltx@ifundefined{DeclareRobustCommand}{%
124 \def\RotCh@temp{\def\EdefRot##1}%
125 }{%
126 \def\RotCh@temp{\DeclareRobustCommand*\EdefRot[1]}%
127 }%
128 }{%
129 \def\RotCh@temp{\protected\def\EdefRot##1}%
130 }
131 \RotCh@temp{%
132 \RotCh@GetNumber{#1}%
133 \ltx@ifundefined{RotCh@rot@romannumeral\RotCh@number}{%
134 \PackageError{rotchiffre}{%
135 Unknown chiffre ROT\RotCh@number
136 }\@ehc
137 \EdefSanitize
138 }{%
139 \RotCh@rot
140 }%
141 }

```

`\RotCh@GetNumber` If ε -TeX is active, then the chiffre number can be an expression supported by `\numexpr`.

```

142 \ltx@ifundefined{numexpr}{%
143 \def\RotCh@GetNumber#1{%
144 \edef\RotCh@number{\number#1}%
145 }%
146 }{%
147 \def\RotCh@GetNumber#1{%
148 \edef\RotCh@number{\the\numexpr#1\relax}%
149 }%
150 }

```

2.5 Set \lccode on a range of characters

```

\RotCh@count
151 \countdef\RotCh@count=255 %

```

```

\RotCh@count@end
152 \countdef\RotCh@count@end=2 %

```

```

RotCh@RangeIgnore
153 \def\RotCh@RangeIgnore{%
154 \RotCh@loop{%
155 \lccode\RotCh@count=\ltx@zero
156 }%
157 }

```

```

\RotCh@RangeSet
158 \ltx@ifundefined{numexpr}{%
159   \countdef\RotCh@count@temp=4 %
160   \def\RotCh@RangeSet#1{%
161     \RotCh@loop{%
162       \RotCh@count@temp=\RotCh@count
163       \advance\RotCh@count@temp #1 %
164       \lccode\RotCh@count=\RotCh@count@temp
165     }%
166   }%
167 }{%
168   \def\RotCh@RangeSet#1{%
169     \RotCh@loop{%
170       \lccode\RotCh@count=\numexpr\RotCh@count#1\relax
171     }%
172   }%
173 }

```

\RotCh@loop

```

174 \def\RotCh@loop#1#2#3{%
175   \RotCh@count=#2 %
176   \RotCh@count@end=#3 %
177   \def\RotCh@action{#1}%
178   \RotCh@@loop
179 }%

```

RotCh@@loop

```

180 \def\RotCh@@loop{%
181   \RotCh@action
182   \ifnum\RotCh@count<\RotCh@count@end
183     \advance\RotCh@count\ltx@one
184     \expandafter\RotCh@@loop
185   \fi
186 }

```

2.6 Chiffres

2.6.1 ROT13

\RotCh@rot@xiii

```

187 \def\RotCh@rot@xiii{%
188   \RotCh@RangeIgnore{0}{64}%
189   \RotCh@RangeSet{+13}{65}{77}%
190   \RotCh@RangeSet{-13}{78}{90}%
191   \RotCh@RangeIgnore{91}{96}%
192   \RotCh@RangeSet{+13}{97}{109}%
193   \RotCh@RangeSet{-13}{110}{122}%
194   \RotCh@RangeIgnore{123}{255}%
195 }

```

2.6.2 ROT5

\RotCh@rot@v

```

196 \def\RotCh@rot@v{%
197   \RotCh@RangeIgnore{0}{47}%
198   \RotCh@RangeSet{+5}{48}{52}%
199   \RotCh@RangeSet{-5}{53}{57}%
200   \RotCh@RangeIgnore{58}{255}%
201 }

```

2.6.3 ROT18

```
\RotCh@rot@xviii
202 \def\RotCh@rot@xviii{%
203 \RotCh@RangeIgnore{0}{47}%
204 \RotCh@RangeSet{+25}{48}{57}%
205 \RotCh@RangeIgnore{58}{64}%
206 \RotCh@RangeSet{+18}{65}{72}%
207 \RotCh@RangeSet{-25}{73}{82}%
208 \RotCh@RangeSet{-18}{83}{90}%
209 \RotCh@RangeIgnore{91}{96}%
210 \RotCh@RangeSet{+13}{97}{109}%
211 \RotCh@RangeSet{-13}{110}{122}%
212 \RotCh@RangeIgnore{123}{255}%
213 }
```

2.6.4 ROT47

```
\RotCh@rot@xlvii
214 \def\RotCh@rot@xlvii{%
215 \RotCh@RangeIgnore{0}{32}%
216 \RotCh@RangeSet{+47}{33}{79}%
217 \RotCh@RangeSet{-47}{80}{126}%
218 \RotCh@RangeIgnore{127}{255}%
219 }
```

2.7 \RotCh@rot with big char support

Some modern \TeX engines support characters with more than eight bits (codes greater as 255). Lua \TeX and Xe \TeX are detected by the caret notation that is extended by these engines.

```
220 \begingroup
221 \catcode0=9 %
222 \catcode'\^=7 %
223 \catcode'\^^=12 %
224 \def\x{^^^0000}%
225 \expandafter\endgroup
226 \ifx\x\ltx@empty
```

```
\RotCh@toks
227 \toksdef\RotCh@toks=0 %
```

```
\RotCh@rot
228 \long\def\RotCh@rot#1#2{%
229 \EdefSanitize1{#2}%
230 \begingroup
231 \csname RotCh@rot@romannumeral\RotCh@number\endcsname
232 \RotCh@toks={}%
233 \expandafter\RotCh@SplitSpace#1 \@nil
234 \expandafter\endgroup
235 \expandafter\def\expandafter#1\expandafter{%
236 \the\RotCh@toks
237 }%
238 }%
```

```
\RotCh@SplitSpace
239 \def\RotCh@temp#1{%
240 \def\RotCh@SplitSpace##1 ##2\@nil{%
241 \RotCh@Add##1\relax
242 \ifx\relax##2\relax
243 \expandafter\ltx@gobble
```

```

244     \else
245     \RotCh@toks\expandafter{\the\RotCh@toks#1}%
246     \expandafter\ltx@firstofone
247     \fi
248     {%
249     \RotCh@SplitSpace##2\@nil
250     }%
251 }%
252 }%
253 \RotCh@temp{ }%

```

\RotCh@Add

```

254 \def\RotCh@Add#1{%
255   \ifx#1\relax
256   \else
257   \ifnum'#1>126 %
258   \RotCh@toks\expandafter{\the\RotCh@toks#1}%
259   \else
260   \lowercase{%
261     \RotCh@toks\expandafter{\the\RotCh@toks#1}%
262   }%
263   \fi
264   \expandafter\RotCh@Add
265   \fi
266 }%
267 \else

```

2.8 \RotCh@rot without big char support

\RotCh@rot

```

268 \long\def\RotCh@rot#1#2{%
269   \EdefSanitize#1{#2}%
270   \begingroup
271   \csname RotCh@rot@\romannumeral\RotCh@number\endcsname
272   \lowercase\expandafter{\expandafter\endgroup
273     \expandafter\def\expandafter#1\expandafter{#1}%
274   }%
275 }%
276 \fi
277 \RotCh@AtEnd%
278 \endpackage

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/oberdiek/rotchiffre.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 T_EX Files” ([CTAN:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

¹[CTAN:pkg/rotchiffre](#)

3.2 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
```

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 rotchiffre.dtx
```

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

```
rotchiffre.sty → tex/generic/oberdiek/rotchiffre.sty
rotchiffre.pdf → doc/latex/oberdiek/rotchiffre.pdf
rotchiffre.dtx → source/latex/oberdiek/rotchiffre.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 `mktextlsr`.

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{rotchiffre.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 rotchiffre.dtx
makeindex -s gind.ist rotchiffre.idx
pdflatex rotchiffre.dtx
makeindex -s gind.ist rotchiffre.idx
pdflatex rotchiffre.dtx
```

4 References

- [1] Stephan Hennig et. al.: *fontspec: no ligatures with Times New Roman*; newsgroup `comp.text.tex`, [news:4cdbed27\\$0\\$6765\\$9b4e6d93@newsspool3.arcor-online.net](mailto:news:4cdbed27$0$6765$9b4e6d93@newsspool3.arcor-online.net), 2010-11-11.
https://groups.google.com/group/comp.text.tex/browse_thread/thread/6266f98e998ce333/d7b32e9dcc610c87
- [2] Stephan Hennig: *Re: fontspec: no ligatures with Times New Roman*; newsgroup `comp.text.tex`, [news:4cdc2abe\\$0\\$6762\\$9b4e6d93@newsspool3.arcor-online.net](mailto:news:4cdc2abe$0$6762$9b4e6d93@newsspool3.arcor-online.net), 2010-11-11.
<https://groups.google.com/group/comp.text.tex/msg/d7b32e9dcc610c87>
- [3] Robin Fairbairns: *Re: fontspec: no ligatures with Times New Roman*; newsgroup `comp.text.tex`, news:qf4obmua0v.fsf@sxp10.cl.cam.ac.uk, 2010-11-12.
<https://groups.google.com/group/comp.text.tex/msg/7c03e91407144704>
- [4] Wikipedia/German: *ROT13*; 2010-10-26.
<https://de.wikipedia.org/wiki/ROT13>
- [5] Wikipedia/English: *ROT13*; 2010-11-11.
<https://en.wikipedia.org/wiki/ROT13>
- [6] Computerfreak/German: *ROT-18*; 2010-04-12.
<http://www.compufreak.info/2010/04/12/rot-18/>
- [7] Wikipedia/English: *The quick brown fox jumps over the lazy dog*; 2010-11-09.
https://en.wikipedia.org/wiki/The_quick_brown_fox_jumps_over_the_lazy_dog

5 History

[2010/11/12 v1.0]

- First version.

[2016/05/16 v1.1]

- 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.

Symbols	C
<code>\@PackageError</code> 134	<code>\catcode</code> 2, 3, 5, 6, 7,
<code>\@ehc</code> 136	8, 9, 10, 11, 12, 13, 33, 34, 36,
<code>\@nil</code> 233, 240, 249	37, 38, 39, 40, 41, 42, 43, 44, 45,
<code>\@undefined</code> 58	46, 47, 48, 49, 69, 70, 72, 73, 74,
<code>\^</code> 222, 223	78, 79, 80, 81, 82, 83, 84, 87, 88,
	90, 91, 92, 93, 97, 99, 221, 222, 223
A	
<code>\advance</code> 163, 183	<code>\countdef</code> 151, 152, 159
<code>\aftergroup</code> 29	<code>\csname</code> 14, 21, 50, 66, 76, 113, 231, 271

D	
<code>\DeclareRobustCommand</code>	126
E	
<code>\EdefRot</code>	2, 122
<code>\EdefSanitize</code>	137, 229, 269
<code>\empty</code>	17, 18
<code>\endcsname</code>	
..	14, 21, 50, 66, 76, 113, 231, 271
<code>\endinput</code>	29, 111
<code>\endlinechar</code>	4, 35, 71, 77, 89
I	
<code>\ifnum</code>	182, 257
<code>\ifx</code>	15,
18, 21, 50, 58, 61, 113, 226, 242, 255	
<code>\immediate</code>	23, 52
<code>\input</code>	114, 115, 116
L	
<code>\lccode</code>	155, 164, 170
<code>\lowercase</code>	260, 272
<code>\ltx@empty</code>	226
<code>\ltx@firstofone</code>	246
<code>\ltx@gobble</code>	243
<code>\ltx@ifundefined</code> 122, 123, 133, 142, 158	
<code>\ltx@one</code>	183
<code>\ltx@zero</code>	155
N	
<code>\number</code>	144
<code>\numexpr</code>	148, 170
P	
<code>\PackageInfo</code>	26
<code>\protected</code>	129
<code>\ProvidesPackage</code>	19, 67
R	
<code>\RequirePackage</code>	118, 119, 120
<code>\romannumeral</code>	133, 231, 271
<code>\RotCh@loop</code>	178, 180, 180, 184
<code>\RotCh@action</code>	177, 181
<code>\RotCh@Add</code>	241, 254
<code>\RotCh@AtEnd</code>	95, 96, 111, 277
<code>\RotCh@count</code>	151,
155, 162, 164, 170, 175, 182, 183	
<code>\RotCh@count@end</code>	152, 176, 182
<code>\RotCh@count@temp</code> .	159, 162, 163, 164
<code>\RotCh@GetNumber</code>	132, 142
<code>\RotCh@loop</code>	154, 161, 169, 174
<code>\RotCh@number</code>	
.....	133, 135, 144, 148, 231, 271
<code>\RotCh@RangeIgnore</code>	
.	153, 153, 188, 191, 194, 197,
200, 203, 205, 209, 212, 215, 218	
<code>\RotCh@RangeSet</code>	158, 189,
190, 192, 193, 198, 199, 204,	
206, 207, 208, 210, 211, 216, 217	
<code>\RotCh@rot</code>	139, 228, 268
<code>\RotCh@rot@v</code>	196
<code>\RotCh@rot@xiii</code>	187
<code>\RotCh@rot@xlvii</code>	214
<code>\RotCh@rot@xviii</code>	202
<code>\RotCh@SplitSpace</code>	233, 239
<code>\RotCh@temp</code> 124, 126, 129, 131, 239, 253	
<code>\RotCh@toks</code> 227, 232, 236, 245, 258, 261	
T	
<code>\the</code>	77, 78, 79, 80, 81, 82,
83, 84, 97, 148, 236, 245, 258, 261	
<code>\TMP@EnsureCode</code> .	94, 101, 102, 103,
104, 105, 106, 107, 108, 109, 110	
<code>\toksdef</code>	227
W	
<code>\write</code>	23, 52
X	
<code>\x</code>	14, 15, 18, 22,
26, 28, 51, 56, 66, 75, 87, 224, 226	