

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	5
2.4	\EdefRot as robust macro	5
2.5	Set \lccode on a range of characters	6
2.6	Chiffres	7
2.6.1	ROT13	7
2.6.2	ROT5	7
2.6.3	ROT18	7
2.6.4	ROT47	7
2.7	\RotCh@rot with big char support	8
2.8	\RotCh@rot without big char support	9
3	Test	9
3.1	Catcode checks for loading	9
3.2	Macro tests	11
3.2.1	Preamble	11
3.2.2	ROT13	13
3.2.3	ROT5	14
3.2.4	ROT18	14
3.2.5	ROT47	14
3.2.6	Big chars	15
4	Installation	15
4.1	Download	15
4.2	Bundle installation	16
4.3	Package installation	16
4.4	Refresh file name databases	16
4.5	Some details for the interested	16

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

5	References	17
6	History	18
	[2010/11/12 v1.0]	18
	[2016/05/16 v1.1]	18

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 $\text{T}_{\text{E}}\text{X}$ 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 `ini- $\text{T}_{\text{E}}\text{X}$` . 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 exeption of the space token that has character code 32 (0x20) and catcode 10 (space). This follows $\text{T}_{\text{E}}\text{X}$'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.boreqvr ng tbbtyrznvy.pbz>
```

2 Implementation

```
1 \<package>
```

2.1 Reload check and package identification

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

```
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 chiffré 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 T_EX engines support characters with more than eight bits (codes greater as 255). LuaT_EX and X_YT_EX 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     \EdefSanitize#1{#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 \</package>

```

3 Test

3.1 Catcode checks for loading

```

279 \<{*test1}

280 \catcode`\{=1 %
281 \catcode`\}=2 %
282 \catcode`\#=6 %
283 \catcode`\@=11 %
284 \expandafter\ifx\csname count@\endcsname\relax
285 \countdef\count@=255 %
286 \fi
287 \expandafter\ifx\csname @gobble\endcsname\relax
288 \long\def\@gobble#1{%
289 \fi
290 \expandafter\ifx\csname @firstofone\endcsname\relax
291 \long\def\@firstofone#1{#1}%
292 \fi
293 \expandafter\ifx\csname loop\endcsname\relax
294 \expandafter\@firstofone
295 \else
296 \expandafter\@gobble
297 \fi
298 {%
299 \def\loop#1\repeat{%
300 \def\body{#1}%
301 \iterate
302 }%
303 \def\iterate{%
304 \body
305 \let\next\iterate
306 \else
307 \let\next\relax
308 \fi
309 \next
310 }%

```

```

311 \let\repeat=\fi
312 }%
313 \def\RestoreCatcodes{}
314 \count@=0 %
315 \loop
316 \edef\RestoreCatcodes{%
317 \RestoreCatcodes
318 \catcode\the\count@=\the\catcode\count@\relax
319 }%
320 \ifnum\count@<255 %
321 \advance\count@ 1 %
322 \repeat
323
324 \def\RangeCatcodeInvalid#1#2{%
325 \count@=#1\relax
326 \loop
327 \catcode\count@=15 %
328 \ifnum\count@<#2\relax
329 \advance\count@ 1 %
330 \repeat
331 }
332 \def\RangeCatcodeCheck#1#2#3{%
333 \count@=#1\relax
334 \loop
335 \ifnum#3=\catcode\count@
336 \else
337 \errmessage{%
338 Character \the\count@\space
339 with wrong catcode \the\catcode\count@\space
340 instead of \number#3%
341 }%
342 \fi
343 \ifnum\count@<#2\relax
344 \advance\count@ 1 %
345 \repeat
346 }
347 \def\space{ }
348 \expandafter\ifx\csname LoadCommand\endcsname\relax
349 \def\LoadCommand{\input rotchiffre.sty\relax}%
350 \fi
351 \def\Test{%
352 \RangeCatcodeInvalid{0}{47}%
353 \RangeCatcodeInvalid{58}{64}%
354 \RangeCatcodeInvalid{91}{96}%
355 \RangeCatcodeInvalid{123}{255}%
356 \catcode`\@=12 %
357 \catcode`\=\0 %
358 \catcode`\%=14 %
359 \LoadCommand
360 \RangeCatcodeCheck{0}{36}{15}%
361 \RangeCatcodeCheck{37}{37}{14}%
362 \RangeCatcodeCheck{38}{47}{15}%
363 \RangeCatcodeCheck{48}{57}{12}%
364 \RangeCatcodeCheck{58}{63}{15}%
365 \RangeCatcodeCheck{64}{64}{12}%
366 \RangeCatcodeCheck{65}{90}{11}%
367 \RangeCatcodeCheck{91}{91}{15}%
368 \RangeCatcodeCheck{92}{92}{0}%

```

```

369 \RangeCatcodeCheck{93}{96}{15}%
370 \RangeCatcodeCheck{97}{122}{11}%
371 \RangeCatcodeCheck{123}{255}{15}%
372 \RestoreCatcodes
373 }
374 \Test
375 \csname @@end\endcsname
376 \end
377 </test1>

```

3.2 Macro tests

3.2.1 Preamble

```

378 \*test2)
379 \catcode`\{=1 %
380 \catcode`\}=2 %
381 \catcode`\#=6 %
382 \catcode`\^=7 %
383 \font\rmfont=ec-lmtt10\relax
384 \rmfont
385 \showboxbreadth=10000 %
386 \showboxdepth=10000 %
387 \errorcontextlines=10000
388 \begingroup\expandafter\expandafter\expandafter\endgroup
389 \expandafter\ifx\csname RequirePackage\endcsname\relax
390 \input rotchiffre.sty\relax
391 \else
392 \RequirePackage{rotchiffre}[2016/05/16]%
393 \RequirePackage{ifluatex}[2010/03/01]%
394 \RequirePackage{ifxetex}[2010/09/12]%
395 \fi
396 \catcode`\@=11 %
397 \begingroup\expandafter\expandafter\expandafter\endgroup
398 \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
399 \begingroup\expandafter\expandafter\expandafter\endgroup
400 \expandafter\ifx\csname detokenize\endcsname\relax
401 \def\strip@prefix#1->{%
402 \def\@onelevel@sanitize#1{%
403 \edef#1{%
404 \expandafter\strip@prefix\meaning#1%
405 }%
406 }%
407 \else
408 \def\@onelevel@sanitize#1{%
409 \edef#1{%
410 \detokenize\expandafter{#1}%
411 }%
412 }%
413 \fi
414 \fi
415 \def\msg#{\immediate\write16}
416 \def\empty{}
417 \begingroup
418 \def\x#1{%
419 \def\space{#1}%
420 \def\spacesII{#1#1}%
421 \def\spacesIII{#1#1#1}%
422 \def\spacesIV{#1#1#1#1}%

```

```

423 }%
424 \expandafter\endgroup\x{ }
425 \def\PrintStr#1#2{%
426   \begingroup
427     \@onelevel@sanitize#2%
428     \msg{#1: [#2]}% hash-ok
429   \endgroup
430 }
431 \def\CheckResult{%
432   \PrintStr{Result}\StrResult
433   \ifx\StrExpect\StrResult
434     \msg{==> Ok}%
435   \else
436     \begingroup
437     \edef\x{\endgroup
438       \errmessage{Test failed (\chiffre)!}%
439     }\x
440   \fi
441 }
442 \long\def\test#1#2{%
443   \msg{ }%
444   \begingroup
445     \setbox0=\hbox{%
446       \edef\StrInput{#1}%
447       \@onelevel@sanitize\StrInput
448       \PrintStr{ Input}\StrInput
449       \edef\StrExpect{#2}%
450       \@onelevel@sanitize\StrExpect
451       \PrintStr{Expect}\StrExpect
452       \action{#1}%
453       \CheckResult
454     }%
455     \ifdim\wd0=0pt %
456     \else
457       \showbox0 %
458     \fi
459   \endgroup
460 }
461 \def\cmd#1{%
462   \msg{* CMD: ROT#1}%
463   \def\chiffre{ROT#1}%
464   \def\action{\EdefRot{#1}\StrResult}%
465 }
466 \def\TestIgnore#1{%
467   \test{#1}{#1}%
468 }
469 \begingroup
470   \lccode`\P=``\%
471   \lccode`\B=``\%
472   \lccode`\H=``\#
473 \lowercase{\endgroup
474   \def\PercentChar{P}%
475   \def\BackslashChar{B}%
476   \def\HashChar{H}%
477 }
478 \def\TestI{%
479   \TestIgnore{%
480     \space!"\HashChar$\PercentChar&'()*+,-./%

```

```

481      :;<=>?@%
482      [\BackslashChar]^_`%
483      \string{|\string}\string~%
484    }%
485  }
486 \begingroup
487   \catcode0=12 %
488   \lccode`A=1 %
489   \lccode`B=2 %
490   \lccode`C=3 %
491   \lccode`D=4 %
492   \lccode`E=5 %
493   \lccode`F=6 %
494   \lccode`G=7 %
495   \lccode`H=8 %
496   \lccode`I=9 %
497   \lccode`J=10 %
498   \lccode`K=11 %
499   \lccode`L=12 %
500   \lccode`M=13 %
501   \lccode`N=14 %
502   \lccode`O=15 %
503   \lccode`P=16 %
504   \lccode`Q=17 %
505   \lccode`R=18 %
506   \lccode`S=19 %
507   \lccode`T=20 %
508   \lccode`U=21 %
509   \lccode`V=22 %
510   \lccode`W=23 %
511   \lccode`X=24 %
512   \lccode`Y=25 %
513   \lccode`Z=26 %
514   \lccode`a=27 %
515   \lccode`b=28 %
516   \lccode`c=29 %
517   \lccode`d=30 %
518   \lccode`e=31 %
519   \lccode`f=127 %
520   \lccode`g=128 %
521   \lccode`h=129 %
522   \lccode`y=254 %
523   \lccode`z=255 %
524 \lowercase{\endgroup
525   \def\TestC{%
526     \TestIgnore{%
527       ^^@ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz%
528     }%
529   }%
530   \def\TestZ{%
531     \TestIgnore{^^@}%
532   }%
533 }

```

3.2.2 ROT13

```

534 \cmd{13}
535 \test{%
536   0123456789%

```

```

537  ABCDEFGHIJKLMNOPQRSTUVWXYZ%
538  abcdefghijklmnopqrstuvwxyz%
539  }{%
540  0123456789%
541  NOPQRSTUVWXYZABCDEFGHIJKLM%
542  nopqrstuvwxyzabcdefghijklmnop%
543  }
544  \TestI
545  \TestC
546  \test{}{}
547  \test{A}{N}
548  \test{N}{A}
549  \test{ }{ }
550  \test{0a}{0n}
551  \test{\spacesIV}{\spacesIV}
552  \test{{}}{}}
553  \test{\par}{\noexpand\cne}

```

3.2.3 ROT5

```

554  \cmd{5}
555  \test{%
556  0123456789%
557  ABCDEFGHIJKLMNOPQRSTUVWXYZ%
558  abcdefghijklmnopqrstuvwxyz%
559  }{%
560  5678901234%
561  ABCDEFGHIJKLMNOPQRSTUVWXYZ%
562  abcdefghijklmnopqrstuvwxyz%
563  }
564  \TestI
565  \TestC

```

3.2.4 ROT18

```

566  \cmd{18}
567  \test{%
568  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789%
569  abcdefghijklmnopqrstuvwxyz%
570  }{%
571  STUVWXYZ0123456789ABCDEFGHIJKLMNQR%
572  nopqrstuvwxyzabcdefghijklmnop%
573  }
574  \TestI
575  \TestC

```

3.2.5 ROT47

```

576  \cmd{47}
577  \test{%
578  !"\HashChar$\PercentChar&'()*+,-./%
579  0123456789%
580  :;<=>?@%
581  ABCDEFGHIJKLMNOPQRSTUVWXYZ%
582  [\BackslashChar]^_`%
583  abcdefghijklmnopqrstuvwxyz%
584  \string{|\string}\string~%
585  }{%
586  PQRSTUVWXYZ%
587  [\BackslashChar]^_`%
588  abcdefghijklmnopqrstuvwxyz%

```

```

589 \string{|\string}\string~%
590 !"\HashChar$\PercentChar&'()*+,-./%
591 0123456789%
592 :;<=>?@%
593 ABCDEFGHIJKLMNOP%
594 }
595 \TestZ
596 \TestC

3.2.6 Big chars

597 \chardef\temp=0 %
598 \begingroup\expandafter\expandafter\expandafter\endgroup
599 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
600 \begingroup\expandafter\expandafter\expandafter\endgroup
601 \expandafter\ifx\csname RequirePackage\endcsname\relax
602 \input ifluatex.sty\relax
603 \else
604 \RequirePackage{ifluatex}[2010/03/01]%
605 \fi
606 \begingroup\expandafter\expandafter\expandafter\endgroup
607 \expandafter\ifx\csname luatexversion\endcsname\relax
608 \else
609 \chardef\temp=1 %
610 \fi
611 \else
612 \chardef\temp=1 %
613 \fi
614 \ifcase\temp
615 \csname @@end\expandafter\endcsname\expandafter\end
616 \fi

617 \msg{* Big chars}
618 \cmd{5}
619 \test{}{}
620 \test{ }{ }
621 \test{ 0 1 }{ 5 6 }
622 \begingroup
623 \lccode`A=300 %
624 \lccode`B=1000 %
625 \lccode`C=10000 %
626 \lowercase{\endgroup
627 \TestIgnore{ABC}%
628 \TestIgnore{x A By zC xAy AxB}%
629 }%

630 \csname @@end\endcsname\end
631 \test2)

```

4 Installation

4.1 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.

¹[CTAN:pkg/rotchiffre](#)

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 rotchiffre.dtx
```

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

<code>rotchiffre.sty</code>	\rightarrow <code>tex/generic/oberdiek/rotchiffre.sty</code>
<code>rotchiffre.pdf</code>	\rightarrow <code>doc/latex/oberdiek/rotchiffre.pdf</code>
<code>test/rotchiffre-test1.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/rotchiffre-test1.tex</code>
<code>test/rotchiffre-test2.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/rotchiffre-test2.tex</code>
<code>rotchiffre.dtx</code>	\rightarrow <code>source/latex/oberdiek/rotchiffre.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 \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

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
```

5 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](https://groups.google.com/group/comp.text.tex/browse_thread/thread/6266f98e998ce333/d7b32e9dcc610c87), 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](https://groups.google.com/group/comp.text.tex/msg/d7b32e9dcc610c87), 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](https://groups.google.com/group/comp.text.tex/msg/7c03e91407144704), 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

6 History

[2010/11/12 v1.0]

- First version.

[2016/05/16 v1.1]

- Documentation updates.

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; plain numbers refer to the code lines where the entry is used.

Symbols			
\#	282, 381, 472	\cne	553
\%	358, 470	\count@	285, 314, 318, 320, 321, 325, 327, 328,
\@	283, 356, 396		329, 333, 335, 338, 339, 343, 344
\@PackageError	134	\countdef	151, 152, 159, 285
\@ehc	136	\csname	14, 21, 50, 66, 76, 113, 231, 271, 284,
\@firstofone	291, 294		287, 290, 293, 348, 375, 389,
\@gobble	288, 296		398, 400, 599, 601, 607, 615, 630
\@nil	233, 240, 249		
\@onelevel@sanitize			
	402, 408, 427, 447, 450		
\@undefined	58		
\\	357, 471		
\{	280, 379		
\}	281, 380		
\^	222, 223, 382		
A		D	
\action	452, 464	\DeclareRobustCommand	126
\advance	163, 183, 321, 329, 344	\detokenize	410
\aftergroup	29		
B		E	
\B	471	\EdefRot	2, 122, 464
\BackslashChar	475, 482, 582, 587	\EdefSanitize	137, 229, 269
\body	300, 304	\empty	17, 18, 416
C		\end	376, 615, 630
\catcode	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 221, 222, 223, 280, 281, 282, 283, 318, 327, 335, 339, 356, 357, 358, 379, 380, 381, 382, 396, 487	\endcsname	14, 21, 50, 66, 76, 113, 231, 271, 284, 287, 290, 293, 348, 375, 389, 398, 400, 599, 601, 607, 615, 630
\chardef	597, 609, 612	\endinput	29, 111
\CheckResult	431, 453	\endlinechar	4, 35, 71, 77, 89
\chiffre	438, 463	\errmessage	337, 438
\cmd	461, 534, 554, 566, 576, 618	\errorcontextlines	387
		F	
		\font	383
		H	
		\H	472
		\HashChar	476, 480, 578, 590
		\hbox	445
		I	
		\ifcase	614
		\ifdim	455
		\ifnum	182, 257, 320, 328, 335, 343
		\ifx	15, 18, 21, 50, 58, 61, 113, 226, 242,

255, 284, 287, 290, 293, 348, 389, 398, 400, 433, 599, 601, 607	
\immediate	23, 52, 415
\input	114, 115, 116, 349, 390, 602
\iterate	301, 303, 305
L	
\lccode	155, 164, 170, 470, 471, 472, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 623, 624, 625
\LoadCommand	349, 359
\loop	299, 315, 326, 334
\lowercase	260, 272, 473, 524, 626
\ltx@empty	226
\ltx@firstofone	246
\ltx@gobble	243
\ltx@ifUndefined	122, 123, 133, 142, 158
\ltx@one	183
\ltx@zero	155
M	
\meaning	404
\msg	415, 428, 434, 443, 462, 617
N	
\next	305, 307, 309
\number	144, 340
\numexpr	148, 170
P	
\P	470
\PackageInfo	26
\par	553
\PercentChar	474, 480, 578, 590
\PrintStr	425, 432, 448, 451
\protected	129
\ProvidesPackage	19, 67
R	
\RangeCatcodeCheck	332, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371
\RangeCatcodeInvalid	324, 352, 353, 354, 355
\repeat	299, 311, 322, 330, 345
\RequirePackage	118, 119, 120, 392, 393, 394, 604
\RestoreCatcodes	313, 316, 317, 372
\rmfont	383, 384
\romannumeral	133, 231, 271
\RotCh@loop	178, 180, 180, 184
\RotCh@action	177, 181
\RotCh@Add	241, 254
\RotCh@AtEnd	95, 96, 111, 277
\RotCh@count	151, 155, 162, 164, 170, 175, 182, 183
\RotCh@count@end	152, 176, 182
\RotCh@count@temp	159, 162, 163, 164
\RotCh@GetNumber	132, 142
\RotCh@loop	154, 161, 169, 174
\RotCh@number	133, 135, 144, 148, 231, 271
\RotCh@RangeIgnore	153, 153, 188, 191, 194, 197, 200, 203, 205, 209, 212, 215, 218
\RotCh@RangeSet	158, 189, 190, 192, 193, 198, 199, 204, 206, 207, 208, 210, 211, 216, 217
\RotCh@rot	139, 228, 268
\RotCh@rot@v	196
\RotCh@rot@xiii	187
\RotCh@rot@xlviii	214
\RotCh@rot@xviii	202
\RotCh@SplitSpace	233, 239
\RotCh@temp	124, 126, 129, 131, 239, 253
\RotCh@toks	227, 232, 236, 245, 258, 261
S	
\setbox	445
\showbox	457
\showboxbreadth	385
\showboxdepth	386
\space	338, 339, 347, 419, 480
\spacesII	420
\spacesIII	421
\spacesIV	422, 551
\StrExpect	433, 449, 450, 451
\StrInput	446, 447, 448
\strip@prefix	401, 404
\StrResult	432, 433, 464
T	
\temp	597, 609, 612, 614
\Test	351, 374
\test	442, 467, 535, 546, 547, 548, 549, 550, 551, 552, 553, 555, 567, 577, 619, 620, 621
\TestC	525, 545, 565, 575, 596
\TestI	478, 544, 564, 574
\TestIgnore	466, 479, 526, 531, 627, 628
\TestZ	530, 595
\the	77, 78, 79, 80, 81, 82, 83, 84, 97, 148, 236, 245, 258, 261, 318, 338, 339
\TMP@EnsureCode	94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
\toksdef	227
W	
\wd	455
\write	23, 52, 415
X	
\x	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 224, 226, 418, 424, 437, 439