

LuaCensor: A package for redacting sensitive information

Elijah Z Granet*

8 April 2024
Version 1.1.1

Contents

1	Very Quick Guide	2
1.1	Purpose	2
1.2	Usage	2
1.3	The warning option	2
1.4	Requirements	3
1.5	Limitations	4
1.6	Demonstration	4
2	More detailed information	5
2.1	Purpose	5
2.2	The censoring mechanism	5
2.3	The accessibility feature	5
2.4	Bugs and development	6
2.5	Licensing	6
2.6	Some useful advice	6
3	Implementation	7
3.1	Dependencies	7
3.2	fonts	8
3.3	Removing pesky environments	9
3.4	The ‘cnsr’ command	10
3.5	the Lua magic	12
3.6	The warning ’ option	15

*e-mail: ezg21@cantab.ac.uk

4	Version History	18
4.1	1.1.1	18
4.2	1.1.0	18
4.3	1.0.0	18

1 Very Quick Guide

1.1 Purpose

This package redacts sensitive information using Lua, and adds accessibility support.

1.2 Usage

The package is called with:

```
\usepackage{luacensor}
```

Sensitive information is enclosed within the command:

```
\cnsr{John Smith}
```

When the outputted document is intended for authorised readers who are supposed to see sensitive information, no further action is needed. When the outputted document is for general audiences, who are *not* supposed to see sensitive information, add the following line to the preamble:

```
\cnsrtrue
```

This activates the censoring globally.

1.3 The **warning** option

For many documents, the presence or absence of redaction in the document will be obvious by black bars in the first page. However, for other documents, particularly long reports or, for example, a court judgment with a title page, it is useful to give an indication. This can be done by calling the package with:

```
\usepackage[warning]{luacensor}
```

This prints automatically prints a warning at the top of the page indicating either that the text is redacted or unredacted. The default ‘safe’ and ‘danger’ warnings, below, are there mostly as a place holder, because it is anticipated that the precise wording of the warning will vary by jurisdiction.

The default danger warning:



This document is NOT redacted. It contains private and confidential personal data, and may NOT be distributed, published, or shown to those without the right to view such information. The publication of the information in this document may constitute a contempt of court, punishable by a term of imprisonment.

The default safe warning:

This document has been altered to remove sensitive personal data. It is cleared for publication and dissemination.

As I noted above, it is unlikely these default options will be suitable for most situations, and for this reason it is easy to change. To alter the text used in the ‘safe’ option, simply use:

```
\renewcommand{\textsafe}{This is the text in safe mode}
```

To change the warning heading option for the ‘danger’ text, renew the command `\warnword` to change the word ‘warning’, renew the command `\dangerblock` to alter or remove the warning triangles. The other commands for more custom changing of the warning are all in the source code and easily altered for even more custom work.

```
\renewcommand{\textwarn}{This is the text in danger mode}
```

To alter the warning triangles,

1.4 Requirements

This package will **only** work in Lua^LTeX. The package works out of the box with a standard TeX distribution, but ideally, I would strongly suggest installing the free (*libre & gratis*) ‘Redacted’ font from Google Fonts, which gives a quite aesthetically pleasing black bar effect.¹

¹Available at: <https://fonts.google.com/specimen/Redacted>; a version is also available at the GitHub repo of this project: <https://github.com/ezgranet/luacensor>

1.5 Limitations

The package is completely effective at censoring text formatted with normal \LaTeX commands, by which I mean that it is impossible for someone to ascertain the original text (or even its precise length) from the outputted PDF when the `\cnsrtrue` option has been activated.

However, math mode is used, it will censor numbers, but not operators or \TeX (as opposed to Unicode) operators. This is probably fine for most instances, but unacceptable where security is of the highest priority, and I would not really recommend using this package to censor highly secret formulæ; the censor package in your \TeX distribution will do a better job of that. The package is set to completely disappear the output (as opposed to black bar over) of the `math`, `align`, `equation`, `tabular`, and a few other environments, as disappearing these environments proved more secure than the piecemeal blacking out I saw. It is probable that there are packages and macros that will break the `cnsr` macro, and therefore, care should be taken to always examine output before public distribution.

Users should also be aware that many \TeX primitives confuse the package, in particular `\vskip`, `\hskip` *etc* that take arguments outside curly braces cause difficulties; the best solution is simply to either use the \LaTeX alternatives (eg, `\hspace`) or enclose the primitives in the `\hddn` command which simply disappears them.

1.6 Demonstration

```
%In the preamble: \usepackage{luacensor}
\begin{quote}
\footnotesize \cnsr{Whereas recognition} of the \cnsrtrue\cnsr{inherent
→ dignity and of the equal and inalienable rights of all members of the
→ human family} is the foundation of freedom, justice and peace in the
→ world,

\cnsr{Whereas disregard and contempt for human rights have resulted in
→ barbarous acts which have outraged the conscience of mankind, and the
→ advent of a world in which human beings shall enjoy freedom of speech
→ and belief and freedom from fear and want has been proclaimed as the
→ highest aspiration of the common people,}
\end{quote}
```

Whereas recognition of the [REDACTED]
[REDACTED] is the foundation of freedom, justice and peace in the world,
[REDACTED]
[REDACTED]
[REDACTED]

2 More detailed information

2.1 Purpose

This package is a relatively lightweight and aesthetically pleasing censorship solution which includes accessibility features to allow screen readers to be aware that content has been redacted.

2.2 The censoring mechanism

The package uses Lua's `toks` filter to replace all UTF8 characters with a single glyph (• in the case of Redacted, and a Unicode black rectangle in the fallback T_EX default font Source Sans Pro). In both font options, these combine visually into a single line (though this can be deconstructed in a text editor).

However, while changing all characters into a single character is effective in *most* cases, this alone would not be sufficiently secure for reliable usage. This is because knowing the length of a censored name could be combined with other information in, for example, a Family Court judgment, to allow for what lawyers call 'jigsaw identification' (*eg*, where there is only one person with an eight letter surname who meets the other details given in the judgment).

Therefore, the package adds an extra layer of security by randomly changing the length of strings during the censorship phase; censored strings can thus be either longer or shorter by a few characters. This means that while the area of the blacked out content will be *approximately* similar to the length of the uncensored string (which means wireframing more or less works), it cannot be used to reverse engineer information about the censored content.

2.3 The accessibility feature

One concern about document redaction is ensuring that visually impaired readers of your document, who use screen reading software to listen to your text, may encounter problems with censored content. If the screen reading software skips over the censored text altogether, it will be a very confusing jump for the visually impaired user. If the screen reading software reads the replacement characters, it will be very annoying for the visually impaired reader to hear, in a censored paragraph, the same character being read out *ad nauseam* (*eg*, 'Asterisk, asterisk, asterisk...').

To overcome this limitation, the package uses the `accsup` package to add an 'actual text' feature which will lead screen readers (and utilities like `pdf to text`) to replace the string of replacement characters with the two words 'TEXT REDACTED'. This also will be encountered by naïve users who try to outdo the package by copying and pasting the black blocks from Adobe™ Acrobat or Reader. (However, because other PDF readers, like Apple's Preview, do not implement accessibility features, this is **not** an additional security feature and is not on its own sufficient to work for redaction; if it were otherwise, the rest of the package would be unnecessary)

2.4 Bugs and development

All bugs, feature requests, or other technical points should be submitted to the package's official Github page.²

2.5 Licensing

The software is free and open-source software licensed under the Latex Public Project Licence, version 1.3c.³

2.6 Some useful advice

This package is really good at some things, but if you find it breaks down on censoring complex L^AT_EX code, the existing `sensor` package on CTAN is excellent (albeit less good with accessibility), and works with non-Lua versions of T_EX. Incidentally, you can use both this package and `sensor` in the same file without trouble; this (*not* a penchant for annoying tech-speak) is why the main command in this package is `cnsr` without vowels.

²<https://github.com/ezgranet/luacensor>

³<https://www.latex-project.org/lppl/>

3 Implementation

```
7 %luacensor.sty
8 \def\luacensorversionnumber{1.1.1}
9 \ProvidesPackage{luacensor}
10 [2024/04/06 \luacensorversionnumber\
11 Redact sensitive information using Lua]
12 % !TeX program = lualatex
13 % !TeX encoding = utf8
14 % This work may be distributed and/or modified under the
15 % conditions of the LaTeX Project Public License, either version 1.3
16 % of this license or (at your option) any later version.
17 % The latest version of this license is in
18 % http://www.latex-project.org/lppl.txt
19 % and version 1.3 or later is part of all distributions of LaTeX
20 % version 2005/12/01 or later.
21 %
22 % This work has the LPPL maintenance status `maintained'.
23 %
24 % The Current Maintainer of this work is Elijah Z Granet
25 %%%%%%%%%%%
26 %%%%%%%%%%%
27 % option (we'll come back
28 % to this later
29 %%%%%%%%%%%
30 %%%%%%%%%%%
31 \newif\ifwarning
32 \warningfalse
33 \DeclareOption{warning}{\warningtrue}
34 \ProcessOptions*
```

3.1 Dependencies

```
35 %%%%%%%%%%%
36 %%%%%%%%%%%
37 % DEPENDENCIES
38 %%%%%%%%%%%
39 %%%%%%%%%%%
40 \RequirePackage{luacode}
41 \RequirePackage{environ}% http://ctan.org/pkg/environ
42
43 \RequirePackage{verbatim}
44 % ^ for the censoring
45 \RequirePackage{accsupp}
```

```

46 %^for accessibility
47 \RequirePackage{fontspec}
48 %^for black lines
49 %in theory, you could do
50 %a lighter version of this
51 %package with just asterisks
52 %or '[TEXT-REDACTED]'
53 %And perhaps that would be better for
54 %the environment with printing
55 %BUT I MADE MY CHOICE!
56 \RequirePackage{xcolor}
57 \RequirePackage{graphicx}

```

3.2 Fonts

```

58 %%%%%%%%%%%
59 %%%%%%%%%%%
60 % FONTS
61 %%%%%%%%%%%
62 %%%%%%%%%%%
63 %%%%%%%%%%%
64 %%%%%%%%%%%
65 % redacted is prettier and free to download
66 %%%%%%%%%%%
67 %%%%%%%%%%%
68 % Strongly recommended
69 %%%%%%%%%%%
70 %%%%%%%%%%%
71 %%%%%%%%%%%
72 %%%%%%%%%%%
73 \IfFontExistsTF{Redacted}{%
74 \newfontface\cnsrfnt[%%%%%%%%%%
75 %the scale is arbitrary, but kind of works
76 %Scale=1.1,
77 %%the below declarations are to prevent warnings about shapes not being
78 ↪ available
79 %WordSpace=0,
80 ItalicFont={Redacted},%
81 BoldItalicFont={Redacted},%
82 BoldFont={Redacted},%
83 SmallCapsFont={Redacted}]{Redacted}
84 \newcommand{\onething}{\cnsrfnt\cdot}
85 \newcommand{\twothings}{\cnsrfnt\cdot\cdot}
86 \newcommand{\donothing}{\cnsrfnt\ }

```



```

86 %%%%%%%%%%
87 %%%%%%%%%%
88 %The little spaces let justification happen
89 %%%%%%%%%%
90 %%%%%%%%%%
91 %%%%%%%%%%
92 %%%%%%%%%%
93 % • chosen as an arbitrary average width
94 %%%%%%%%%%
95 %%%%%%%%%%
96 }{
97 %%%%%%%%%%
98 %%%%%%%%%%
99 % This option works perfectly
100 %fine, it's just less pretty
101 %%but a good fallback because
102 % Source Sans is in TeX dists by default
103 %%%%%%%%%%
104 %%%%%%%%%%
105 \newfontface\cnsrfnt[Scale=1.01,%To allow for separate use of source sans in
↪ text
106 WordSpace=0,%To make it all one black line
107 %the below declarations are to prevent warnings about shapes not being
↪ available
108 ItalicFont={Source Sans Pro Black},BoldItalicFont={Source Sans Pro
↪ Black},BoldFont={Source Sans Pro Black},SmallCapsFont={Source Sans Pro
↪ Black}]{Source Sans Pro Black}
109 %%%%%%%%%%
110 %%%%%%%%%%
111 % Bit of unicode magic below to make the black line effect
112 %%%%%%%%%%
113 %%%%%%%%%%
114 \newcommand{\onething}{\cnsrfnt - }
115 \newcommand{\twothings}{\cnsrfnt - - }
116 \newcommand{\donothing}{ }
117 }

```

3.3 Removing pesky environments

```

118 %%%%%%%%%%
119 %%%%%%%%%%
120 % A neat fallback for disappearing things...
121 %%%%%%%%%%
122 %%%%%%%%%%

```

```

123 %%%%%%%%%%%
124 %%%%%%%%%%%
125 % FULL CREDIT
126 % and FULSOME THANKS
127 % TO TEX.SE USER
128 % Werner for the code below
129 %%%%%%%%%%%
130 %%%%%%%%%%%
131 \makeatletter
132 \newcommand{\voidenvironment}[1]{%
133   \expandafter\providecommand\csname env@#1@save@env\endcsname{%
134   \expandafter\providecommand\csname env@#1@process\endcsname}%
135   \@ifundefined{#1}{\RenewEnviron{#1}}%
136 }
137 \makeatother
138 \newcommand{\hddn}[1]{%
139 \ifcnsr{}\else%
140 #1\fi}
141 \newenvironment*{hidden}{\begin{@empty}
142 }{\end{@empty}}
143 \voidenvironment{hidden}
144
145 %%%%%%%%%%%
146 %%%%%%%%%%%

```

3.4 The ‘cnsr’ command

```

147 %%%%%%%%%%%
148 %%%%%%%%%%%
149 %%%%%%%%%%%
150 %%%%%%%%%%%
151 % the CENSOR COMMAND
152 %%%%%%%%%%%
153
154 \newif\ifcnsr\cnsrfalse
155 \newcommand{\cnsr}[1]{%
156 \ifcnsr{%
157 \voidenvironment{equation*}%
158 \voidenvironment{equation}%
159 \voidenvironment{table}%
160 \voidenvironment{table*}%
161 \voidenvironment{tabular}%
162 \voidenvironment{tabular*}%
163 \voidenvironment{}}%

```

```

164 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
165 % I don't know how many
166 % people use TEX native accent commands
167 % in LuaTEX given that using Unicode is more
168 %people's style. But just in case, because these can lead to stray accent
   ↪ marks floating above censored letters.
169 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
170 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
171 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
172 \renewcommand{\`}[1]{}%
173 \renewcommand{\'}[1]{}%
174 \renewcommand{\^}[1]{}%
175 \renewcommand{\"}[1]{}%
176 \renewcommand{\H}[1]{}%
177 \renewcommand{\~}[1]{}%
178 \renewcommand{\c}[1]{}%
179 \renewcommand{\k}[1]{}%
180 \renewcommand{\l}[1]{}%
181 \renewcommand{\=}[1]{}%
182 \renewcommand{\b}[1]{}%
183 \renewcommand{\.}[1]{}%
184 \renewcommand{\d}[1]{}%
185 \renewcommand{\r}[1]{}%
186 \renewcommand{\u}[1]{}%
187 \renewcommand{\v}[1]{}%
188 \renewcommand{\t}[1]{}%
189 \renewcommand{\o}[1]{}%
190 \renewcommand{\i}[1]{}%
191 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%d
192 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
193 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
194 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
195 % here we have the accsupp magic
196 % this operates by replacing the 'x's
197 % or unicode black squares as the case
198 % may be with an alt text
199 % this serves a dual purpose of both making
200 %pdftotext not break with huge strings of meaningless characters
201 %but more importantly
202 % it means screen readers don't subject
203 % . their users to the meaningless reading out of unicode black squares 50
   ↪ times in a row!
204 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
205 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
206 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
207 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
208 \BeginAccSupp{method=plain,ActualText={TEXT REDACTED}}%

```

```

209 \rndstring{#1}%
210 \EndAccSupp{}}%
211 \else%
212 %%%%%%%%%%%
213 %%%%%%%%%%%
214 % if the conditional is off
215 % the command does absolutely nothing
216 %%%%%%%%%%%
217 %%%%%%%%%%%
218 #1%
219 \fi}
220 %%%%%%%%%%%
221 %%%%%%%%%%%

```

3.5 the Lua magic

```

222 % %%%%%%%%%%%
223 %%%%%%%%%%%
224 % The LUA MAGIC PART
225 %%%%%%%%%%%
226 %%%%%%%%%%%
227 %%%%%%%%%%%
228 %%%%%%%%%%%
229 \begin{luacode}

```

```

230 --fulsome thanks to TeX.SE users Henri Menke and David Carlisle, without whom
231 ↪ none of this would be possible
232 local function rndstring()
233     local toks = token.scan_toks(s)
234     local on = true
235     for n, t in ipairs(toks) do
236         if t.csname == "begin" or t.csname == "end" then
237             on = false
238         end
239         -- The below is necessary as TeX primitives can break the code otherwise
240         ↪ because they do not use brackets
241         if not(on) and t.cmdname == "right_brace" then
242             on = true
243             -- This prevents needless errors about gibberish up commands
244         end
245     end

```

```

244 if on and t.csname == "&" then
245     local letter = token.create'donothing'
246     toks[n] = letter
247
248 elseif on and t.csname == "%" then
249     local letter = token.create'donothing'
250     toks[n] = letter
251
252 elseif on and t.csname == "$" then
253     local letter = token.create'donothing'
254     toks[n] = letter
255
256 elseif on and t.csname == "#" then
257     local letter = token.create'donothing'
258     toks[n] = letter
259
260 elseif on and t.csname == "_" then
261     local letter = token.create'donothing'
262     toks[n] = letter
263
264 elseif on and t.csname == "{" then
265     local letter = token.create'donothing'
266     toks[n] = letter
267
268 elseif on and t.csname == "}" then
269     local letter = token.create'donothing'
270     toks[n] = letter
271
272 elseif on and t.csname == "~" then
273     local letter = token.create'donothing'
274     toks[n] = letter
275
276 elseif on and t.csname == "^" then
277     local letter = token.create'donothing'
278     toks[n] = letter
279 elseif on and t.cmdname == "letter" then
280     -- The below is the randomness part of this, which I admit is fairly
281     ↪ arbitrary, but will more often artificially shorten strings than lengthen
282     ↪ them, as testing found if lengthening was too frequent, it led to really
283     ↪ unsightly long strings.
284     local f = math.random (1,20)
285     if f == 1 then
286         local letter = token.create'donothing'
287         toks[n] = letter
288
289         elseif f == 2 then
290             local letter = token.create'donothing'

```

```

288     toks[n] = letter
289     elseif f == 3 then
290         local letter = token.create'donothing' toks[n] = letter
291     elseif f == 4 then
292         local letter = token.create'twothings'
293     toks[n] = letter
294     elseif f == 5 then
295         local letter = token.create'donothing' toks[n] = letter
296
297     else
298         local letter = token.create'onething'
299     toks[n] = letter
300         end
301         elseif
302         on and t.cmdname == "spacer" then
303             local f = math.random (1,20)
304             if f == 2 then
305                 local letter = token.create'donothing'
306             toks[n] = letter
307                 elseif f == 3 then
308
309                     local letter = token.create'donothing' toks[n] = letter
310             elseif f == 4 then
311                 local letter = token.create'donothing'
312             toks[n] = letter
313                 elseif f == 5 then
314
315
316                     local letter = token.create'twothings' toks[n] = letter
317             elseif f == 6 then
318
319
320                     local letter = token.create'donothing' toks[n] = letter
321                     elseif f == 7 then
322                 local letter = token.create'donothing' toks[n] = letter
323
324
325             else
326                 local letter = token.create'onething'
327             toks[n] = letter
328
329                 end
330
331                 elseif
332                 on and t.cmdname == "other_char" then
333                     local f = math.random (1,20)
334                     if f == 2 then

```

```

335         local letter = token.create'donothing'
336     toks[n] = letter
337         elseif f == 3 then
338
339             local letter = token.create'donothing'   toks[n] = letter
340     elseif f == 4 then
341         local letter = token.create'donothing'
342         toks[n] = letter
343         elseif f == 5 then
344
345             local letter = token.create'twothings'   toks[n] = letter
346     elseif f == 6 then
347
348
349             local letter = token.create'donothing'   toks[n] = letter
350         elseif f == 7 then
351             local letter = token.create'donothing'   toks[n] = letter
352
353
354     else
355         local letter = token.create'onething'
356     toks[n] = letter
357
358         end
359     end
360
361     end
362     --Drop the token in and move on
363     token.put_next(toks)
364 end
365 local lft = lua.get_functions_table()
366 --make a global command
367 lft[#lft + 1] = rndstring
368 token.set_lua("rndstring", #lft, "global")

```

```

369 \end{luacode}

```

3.6 The 'warning' option

```

370 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
371 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

372 % WARNING FUN YAY
373 %%%%%%%%%%%
374 %%%%%%%%%%%
375 %%%%%%%%%%%
376 %%%%%%%%%%%
377 % Definitely this whole section
378 % is there to be user modified, because
379 % depending on language, jurisdiction
380 % type of document etc, everyone will need
381 % a specific warning style. So the important
382 % part of the code here is the
383 % conditional and global [warning]
384 % option, because that's the magic value added
385 %%%%%%%%%%%
386 %%%%%%%%%%%
387 %%%%%%%%%%%
388 %%%%%%%%%%%
389 %%%%%%%%%%%
390 %%%%%%%%%%%
391 % fonts for the warning:
392 % I chose default LaTeX fonts
393 % here to be changed as users wish
394 %%%%%%%%%%%
395 %%%%%%%%%%%
396 \newfontface\wrnstncl{QT Military}
397 \newcommand{\warnword}{WARNING}
398 \newfontface\smb1{DejaVuSans-Bold}
399 \newcommand{\danger}{\smb1 \triangle normalfont}
400 \newcommand{\warnformat}{\sffamily\bfseries \color{red}}
401 \newcommand{\textwarn}{This document is {\underline{NOT}} redacted. It
↪ contains private and confidential personal data, and may {\underline{NOT}}
↪ be distributed, published, or shown to those without the right to view
↪ such information. The publication of the information in this document may
↪ constitute a contempt of court, punishable by a term of imprisonment.}
402 \newcommand{\textsafe}{This document has been altered to remove sensitive
↪ personal data. It is cleared for publication and dissemination.
403 }
404 \definecolor{darkgreen}{rgb}{0.0, 0.2, 0.13}
405 \definecolor{darkspringgreen}{rgb}{0.09, 0.45, 0.27}
406 \definecolor{forestgreen}{rgb}{0.13, 0.55, 0.13}
407 \newcommand{\dquad}{\danger\danger\danger\danger}
408 \newcommand{\dangersign}[1]{\scalebox{2}{\huge\danger}}
409 \newcommand{\dangerblock}{\scalebox{2}{\huge\danger\quad\danger\quad\danger}}
410 \newcommand{\warnblock}{\Large\wrnstncl\warnword\quad\warnword\quad\warnword}
411 \newcommand{\tworules}{\hrule width \hsize height .7pt\vskip2pt\hrule width
↪ \hsize height .7pt}
412 \newcommand{\allwarning}{\dangerblock\\warnblock\\%

```



```

413 \normalfont\smallskip\warnformat\textwarn }
414 \newcommand{\confwarning}{%
415 %%%%%%%%%%%
416 %%%%%%%%%%%
417 % The warning option
418 %%%%%%%%%%%
419 %%%%%%%%%%%
420 \ifwarning
421 \ifcnsr
422 %%%%%%%%%%%
423 %%%%%%%%%%%
424 % a note saying document is redacted
425 %%%%%%%%%%%
426 %%%%%%%%%%%
427 \begin{center}
428 \color{forestgreen}
429 \tworules\vskip5pt
430 \normalsize\normalfont\sffamily\bfseries\textsafe
431 \vskip5pt\tworules
432 \end{center}
433 \else
434 %%%%%%%%%%%
435 %%%%%%%%%%%
436 % The WARNING for un-redacted docs
437 %%%%%%%%%%%
438 %%%%%%%%%%%
439 \begin{center}\color{red}\tworules\vskip 5pt\allwarning
440 \vskip5pt\tworules%
441 \end{center}%
442 \fi%
443 \else\fi}
444 %%%%%%%%%%%
445 %%%%%%%%%%%
446 % Allow \maketitle
447 % on same page
448 % yay
449 %
450 %%%%%%%%%%%
451 %%%%%%%%%%%
452 \ifwarning\let\oldmaketitle\maketitle\renewcommand{\maketitle}%
453 %
454 {{\let\newpage\relax\maketitle}}\else\fi
455 %%%%%%%%%%%
456 %%%%%%%%%%%
457 % print the warning at the start of the document
458 %%%%%%%%%%%
459 %%%%%%%%%%%

```

4 Version History

4.1 1.1.1

6 April 2024: Fixed an issue with OTF for dejavusans not loading by using the postscript name.

4.2 1.1.0

22 February 2022: Added the ``warning'` option and fixed a few errors in the code resulting from \TeX primitives causing issues.

4.3 1.0.0

18 February 2022: Package creation