`LuaCensor`: A package for redacting sensitive information

Elijah Z Granet*

23 February 2022
`Version 1.1.0`

Contents

---

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

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:

⚠ ⚠ ⚠

WARNING WARNING 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 LuaLaTeX. 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 æsthetically pleasing black bar effect.[1]

---

[1] Available at: `https://fonts.google.com/specimen/Redacted`; a versioin 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] is the foundation of freedom, justice and peace in the world,

[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 Luas'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 TEX 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 `pdftotext`) 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 1.3*c*.[3]

2.6 Some useful advice

This package is really good at some things, but if you find it breaks down on censoring complex LaTeX code, the existing `censor` package on CTAN is excellent (albeit less good with accessibility), and works with non-`Lua` versions of TeX. Incidentally, you can use both this package and `censor` 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

```
%luacensor.sty
%luacensor.sty
\def\luacensorversionnumber{1.1.0}
\ProvidesPackage{luacensor}
[2022/02/22 \luacensorversionnumber\
 Redact sensitive information using Lua]
% !TeX program = lualatex
% !TeX encoding = utf8
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%   http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status `maintained'.
%
% The Current Maintainer of this work is Elijah Z Granet
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% option (we'll come back
% to this later
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newif\ifwarning
\warningfalse
\DeclareOption{warning}{\warningtrue}
\ProcessOptions*
```

3.1 Dependencies

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DEPENDENCIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\RequirePackage{luacode}
\RequirePackage{environ}% http://ctan.org/pkg/environ

\RequirePackage{verbatim}
% ^ for the censoring
```

```
\RequirePackage{accsupp}
%^for accessibility
\RequirePackage{fontspec}
%^for black lines
%in theory, you could do
%a lighter version of this
%package with just asterisks
%or `[TEXT-REDACTED]'
%And perhaps that would be better for
%the environment with printing
%BUT I MADE MY CHOICE!
\RequirePackage{xcolor}
\RequirePackage{graphicx}
```

3.2 fonts

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FONTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% redacted is prettier and free to download
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Strongly recommended
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\IfFontExistsTF{Redacted}{%
\newfontface\cnsrfnt[%%%%%%
%the scale is arbitrary, but kind of works
%Scale=1.1,
%%the below declarations are to prevent warnings about shapes not being available
%WordSpace=0,
ItalicFont={Redacted},%
BoldItalicFont={Redacted},%
BoldFont={Redacted},%
SmallCapsFont={Redacted}]{Redacted}
\newcommand{\onething}{\cnsrfnt\ • }
\newcommand{\twothings}{\cnsrfnt\ • •}
\newcommand{\donothing}{\cnsrfnt\ }
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The little spaces let justification happen
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% • chosen as an arbitrary average width
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
}{
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This option works perfectly
%fine, it's just less pretty
%%but a good fallback because
% Source Sans is in TeX dists by default
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newfontface\cnsrfnt[Scale=1.01,%To allow for separate use of source sans in
↪  text
WordSpace=0,%To make it all one black line
%the below declarations are to prevent warnings about shapes not being
↪  available
ItalicFont={Source Sans Pro Black},BoldItalicFont={Source Sans Pro
↪  Black},BoldFont={Source Sans Pro Black},SmallCapsFont={Source Sans Pro
↪  Black}]{Source Sans Pro Black}
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bit of unicode magic below to make the black line effect
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand{\onething}{\cnsrfnt ▬ }
\newcommand{\twothings}{\cnsrfnt ▬ ▬ }
\newcommand{\donothing}{ }
}
```

3.3 Removing pesky environments

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A neat fallback for disappearing things…
%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FULL CREDIT
% and FULSOME THANKS
% TO TEX.SE USER
% Werner  for the code below
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\makeatletter
\newcommand{\voidenvironment}[1]{%
  \expandafter\providecommand\csname env@#1@save@env\endcsname{}%
  \expandafter\providecommand\csname env@#1@process\endcsname{}%
  \@ifundefined{#1}{}{\RenewEnviron{#1}{}}%
}
\makeatother
\newcommand{\hddn}[1]{%
\ifcnsr{}\else%
#1\fi}
\newenvironment*{hidden}{\begin{@empty}
}{\end{@empty}}
\voidenvironment{hidden}

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.4 The ‘`cnsr`’ command

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the CENSOR COMMAND
%%%%%%%%%%%%%%%%%%%%%%%%%%%

\newif\ifcnsr\cnsrfalse
\newcommand{\cnsr}[1]{%
\ifcnsr{%
\voidenvironment{equation*}%
\voidenvironment{equation}%
\voidenvironment{table}%
\voidenvironment{table*}%
\voidenvironment{tabular}%
\voidenvironment{tabular*}%
```

```
\voidenvironment{}%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% I don't know how many
% people use TEX native accent commands
% in LuaTEX given that using Unicode is more
%people's style.  But just in case, because these can lead to stray accent marks floating above censored letters.
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\renewcommand{\`}[1]{}%
\renewcommand{\'}[1]{}%
\renewcommand{\^}[1]{}%
\renewcommand{\"}[1]{}%
\renewcommand{\H}[1]{}%
\renewcommand{\~}[1]{}%
\renewcommand{\c}[1]{}%
\renewcommand{\k}[1]{}%
\renewcommand{\l}[1]{}%
\renewcommand{\=}[1]{}%
\renewcommand{\b}[1]{}%
\renewcommand{\.}[1]{}%
\renewcommand{\d}[1]{}%
\renewcommand{\r}[1]{}%
\renewcommand{\u}[1]{}%
\renewcommand{\v}[1]{}%
\renewcommand{\t}[1]{}%
\renewcommand{\o}[1]{}%
\renewcommand{\i}[1]{}%
%%%%%%%%%%%%%%%%%%%%%%%%%%%d
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% here we have the accsupp magic
% this operates by replacing the 'x's
% or unicode black squares as the case
% may be with an alt text
% this serves a dual purpose of both making
%pdftotext not break with huge strings of meaningless characters
%but more importantly
% it means screen readers don't subject
%. their users to the meaningless reading out of unicode black squares 50 times in a row!
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
\BeginAccSupp{method=plain,ActualText={TEXT REDACTED}}%
\rndstring{#1}%
\EndAccSupp{}}%
\else%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% if the conditional is off
% the command does absolutely nothing
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
#1%
\fi}
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.5 the Lua magic

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The LUA MAGIC PART
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{luacode}
```

```
--fulsome thanks to TeX.SE users Henri Menke and David Carlisle, without whom none of this would be possible
local function rndstring()
    local toks = token.scan_toks(s)
        local on = true
for n, t in ipairs(toks) do
    if t.csname == "begin" or t.csname == "end" then
       on = false
-- The below is necessary as TeX primitives can break the code otherwise because they do not use brackets
    end

   if not(on) and t.cmdname == "right_brace" then
     on = true
     -- This prevents needless errors about gibberish up commands
```

```
end
if on and t.csname ==  "&" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "%" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "$" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "#" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "_" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "{" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "}" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "~" then
  local letter = token.create'donothing'
 toks[n] = letter

elseif on and t.csname == "^" then
  local letter = token.create'donothing'
 toks[n] = letter
elseif on and t.cmdname ==  "letter" then
-- The below is the randomness part of this, which I admit is fairly arbitrary, but will more often  artificially shorten strings than lengthen them, as testing found if lengthening was too frequent, it led to really unsightly long strings.
            local f = math.random (1,20)
            if f == 1 then
               local letter = token.create'donothing'
 toks[n] = letter

                    elseif f == 2 then
```

```
              local letter = token.create'donothing'
toks[n] = letter
elseif f == 3 then
  local letter = token.create'donothing'  toks[n] = letter
elseif f == 4 then
              local letter = token.create'twothings'
 toks[n] = letter
  elseif f == 5 then
  local letter = token.create'donothing'  toks[n] = letter

  else
              local letter = token.create'onething'
toks[n] = letter
                   end
                   elseif
           on and t.cmdname == "spacer" then
           local f = math.random (1,20)
                   if f == 2 then
              local letter = token.create'donothing'
toks[n] = letter
                                    elseif f == 3 then

              local letter = token.create'donothing'  toks[n] = letter
elseif f == 4 then
local letter = token.create'donothing'
 toks[n] = letter
                                      elseif f == 5 then


              local letter = token.create'twothings'  toks[n] = letter
elseif f == 6 then


              local letter = token.create'donothing'  toks[n] = letter
                 elseif f == 7 then
  local letter = token.create'donothing'  toks[n] = letter


  else
              local letter = token.create'onething'
toks[n] = letter

                   end

                   elseif
           on and t.cmdname == "other_char" then
           local f = math.random (1,20)
```

```
                    if f == 2 then
               local letter = token.create'donothing'
 toks[n] = letter
                                     elseif f == 3 then

               local letter = token.create'donothing'  toks[n] = letter
 elseif f == 4 then
 local letter = token.create'donothing'
  toks[n] = letter
                                       elseif f == 5 then


               local letter = token.create'twothings'  toks[n] = letter
 elseif f == 6 then


               local letter = token.create'donothing'  toks[n] = letter
                  elseif f == 7 then
   local letter = token.create'donothing'  toks[n] = letter


   else
               local letter = token.create'onething'
 toks[n] = letter

                    end
                    end
    end
 --Drop the token in and move on
        token.put_next(toks)
end
local lft = lua.get_functions_table()
--make a global command
lft[#lft + 1] = rndstring
token.set_lua("rndstring", #lft, "global")
```

```
\end{luacode}
```

3.6 The ‘`warning`’ option

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% WARNING FUN YAY
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definitely this whole section
%is there to be user modified, because
% depending on language, jurisdiction
%type of document etc, everyone will need
%a specific warning style. So the important
% part of the code here is the
% conditional and global [warning]
% option, because that's the magic value added
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fonts for the warning:
%I chose default LaTeX fonts
% here to be changed as users wish
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newfontface\wrnstncl{QT Military}
\newcommand{\warnword}{WARNING}
\newfontface\smbl{Deja Vu Sans Bold}
\newcommand{\danger}{\smbl ⚠\normalfont}
\newcommand{\warnformat}{\sffamily\bfseries \color{red}}
\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.}
\newcommand{\textsafe}{This document has been altered to remove sensitive
 ↪  personal data.  It is cleared for publication and dissemination.
}
\definecolor{darkgreen}{rgb}{0.0, 0.2, 0.13}
\definecolor{darkspringgreen}{rgb}{0.09, 0.45, 0.27}
        \definecolor{forestgreen}{rgb}{0.13, 0.55, 0.13}
\newcommand{\dquad}{\danger\danger\danger\danger}
\newcommand{\dangersign}[1]{\scalebox{2}{\huge\danger}}
\newcommand{\dangerblock}{\scalebox{2}{\huge\danger\quad\danger\quad\danger}}
\newcommand{\warnblock}{{\Large\wrnstncl\warnword\quad\warnword\quad\warnword}}
\newcommand{\tworules}{\hrule width \hsize height .7pt\vskip2pt\hrule width
 ↪  \hsize height .7pt}
```

```latex
\newcommand{\allwarning}{\dangerblock\\\warnblock\\\normalfont\smallskip\warnformat\textwarn}
\newcommand{\confwarning}{%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The warning option
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\ifwarning
\ifcnsr
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% a note saying document is redacted
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{center}
\color{forestgreen}
\tworules\vskip5pt
\normalsize\normalfont\sffamily\bfseries\textsafe
\vskip5pt\tworules
\end{center}
\else
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The WARNING for un-redacted docs
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{center}\color{red}\tworules\vskip 5pt\allwarning
\vskip5pt\tworules%
\end{center}%
\fi%
\else\fi}
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Allow \maketitle
% on same page
% yay
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\ifwarning\let\oldmaketitle\maketitle\renewcommand{\maketitle}{{\let\newpage\relax\maketitle}}\else\fi
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% print the warning at the start of the document
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
\AtBeginDocument{\confwarning}
```

4 Version History

4.1 1.1.0

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

4.2 1.0.0

18 February 2022: Package creation