The lua-visual-debug package (V1.1)

Patrick Gundlach, Udi Fogiel

April 24, 2026

Contents

1 About **1**

2 How to use **1**

3 A LaTeX example **1**

4 A plain TeX example **3**

5 How to interpret the markers **5**

6 Configuration **6**

7 Copying **8**

1 About

This package aids debugging your TeX and LaTeX document by drawing rectangles around boxes and rules where glue is inserted. Other items are marked as well: kerns, hyphenation points and penalties.

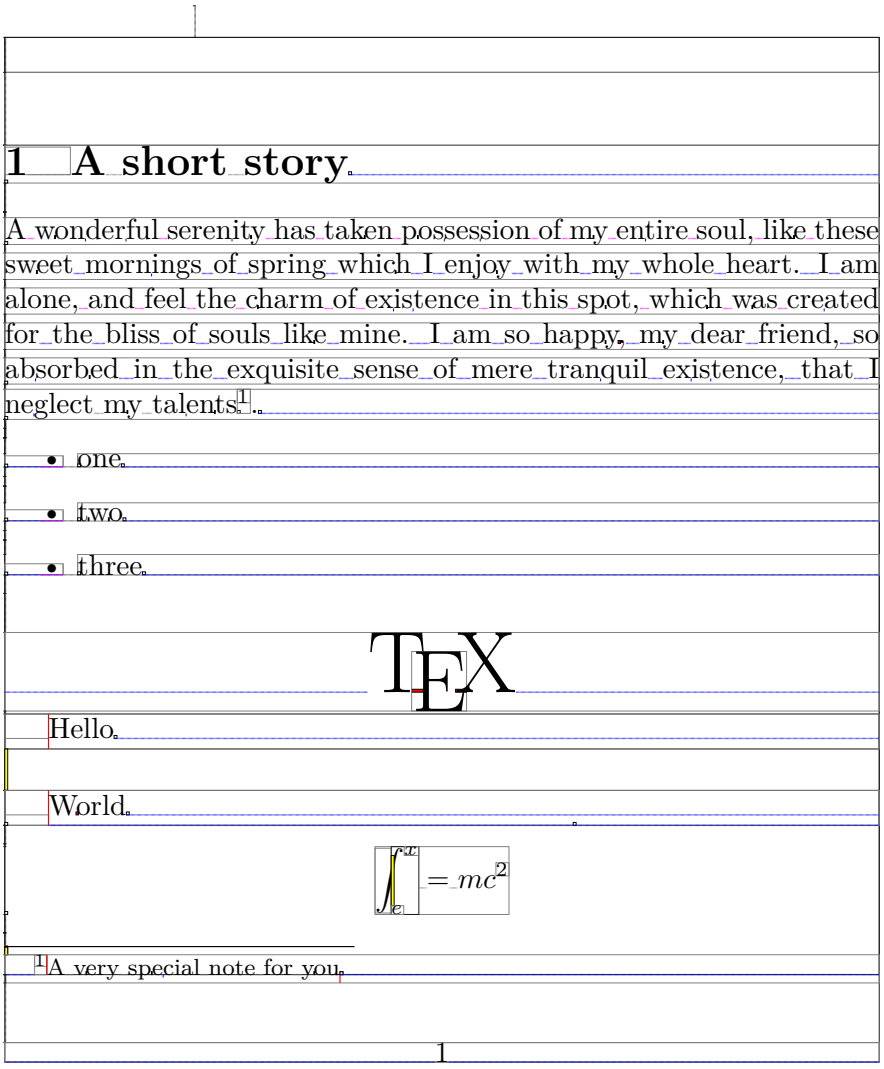
2 How to use

When you load the package `lua-visual-debug` in your LuaLaTeX document (or use `\input lua-visual-debug.sty` in plain TeX, or `\load[lua-visual-debug]` in OPTeX), LuaTeX will highlight boxes, penalties, glues and kerns in the PDF. This package requires you to process the document with LuaTeX (plain, LaTeX or OPTeX formats).

3 A LaTeX example

```
\documentclass{article}
\usepackage{lua-visual-debug}

\setlength\textwidth{300pt}
\setlength\textheight{10cm}
\pagewidth\dimexpr\textwidth+.2in
\pageheight\dimexpr\textheight+1.25in
\pdfvariable vorigin 0in
\oddsidemargin-.9in
\topmargin.15in

\begin{document}

\section{A short story}

A wonderful serenity has taken possession of my entire soul, like these sweet
mornings of spring which I enjoy with my whole heart. I am alone, and feel
the charm of existence in this spot, which was created for the bliss of souls
like mine. I am so happy, my dear friend, so absorbed in the exquisite sense
of mere tranquil existence, that I neglect my talents\footnote{A very special
    ↪ note for you}.

\begin{itemize}
  \item one
  \item two
  \item three
\end{itemize}

\bgroup\fontsize{30}{34}\selectfont
\centerline{\TeX}
\egroup

\vbox{\strut Hello}\kern .5cm\vbox{\strut World}

\[ \int_e^x=mc^2 \]

\end{document}
```

yields



4 A plain TEX example

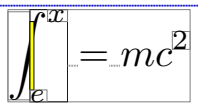
```
\input lua-visual-debug.sty
\hsize 3in \vsize 3in
\pagewidth 3.2in \pageheight 4in
\hoffset -.9in \voffset -.5in
\centerline{A centered line with \TeX}
\vskip .5in
A wonderful serenity has taken possession of my entire soul, like these sweet
mornings of spring which I enjoy with my whole heart. I am alone, and feel
the charm of existence in this spot, which was created for the bliss of souls
like mine. I am so happy, my dear friend, so absorbed in the exquisite sense
of mere tranquil existence, that I neglect my talents.
$$\int_e^x=mc^2$$
```

```
\bye
```

yields

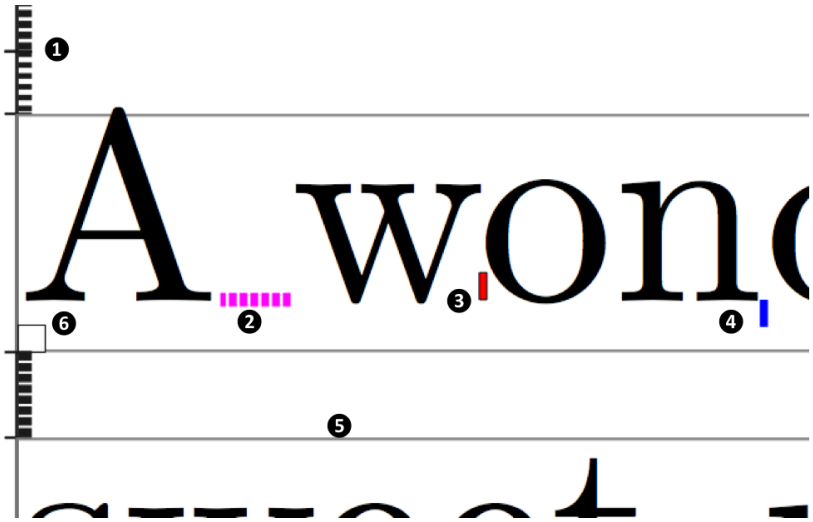
A centered line with TEX

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents.



1

5 How to interpret the markers



1. A vertical glue. Beginning and end are marked with a small tick. At the mark 1, two vertical glues are connected.
2. A horizontal glue. Blue dashed lines represent stretched glues, magenta lines represent shrunk glues, gray at their natural width.
3. A negative kern. Positive kerns are yellow.
4. A possible hyphenation point.
5. Horizontal and vertical boxes are drawn with a border.
6. Penalties are marked with a square. A penalty of 10,000 is marked with a blank square, a penalty less than 10,000 is filled with a gray square (that will improve in the future, currently it is grayness of penalty / 10000).

A strut box (zero width box) is marked with a red rule:

6 Configuration

If the `luakeyval` package is installed, then you can use the `\lvdset` macro modifies the markers described in Section 5. It accepts a list of space-separated `key/val` pairs. Most keys accept nested `key/val` pairs enclosed in curly braces. PDF operators (for `color`, `negative_color`, and `opacity`) must be enclosed in curly braces, e.g., `\lvdset{glyph={color={1 0 0 RG}}}`.

Key	Sub-key	Default	Description
`hlist`	`show`	`true`	Whether to mark hlists
	`color`	`0.5 G`	PDF stroking color operator
	`width`	`0.1`	Line width in bp units
`vlist`	`show`	`true`	Whether to mark vlists
	`color`	`0.1 G`	PDF stroking color operator
	`width`	`0.1`	Line width in bp units
`rule`	`show`	`true`	Whether to mark rules
	`color`	`1 0 0 RG`	PDF stroking color operator
	`width`	`0.4`	Line width in bp units
`disc`	`show`	`true`	Whether to mark discretionaries
	`color`	`0 0 1 RG`	PDF stroking color operator
	`width`	`0.3`	Line width in bp units
`glue`	`show`	`true`	Whether to mark glue
`kern`	`show`	`true`	Whether to mark kerns
	`color`	`1 1 0 rg`	PDF color for positive kerns (stroke and fill)
	`negative_color`	`1 0 0 rg`	PDF color for negative kerns (stroke and fill)
	`width`	`1`	Line width in bp units
`penalty`	`show`	`true`	Whether to mark penalties
	`colorfunc`	(see below)	Lua function that accepts the penalty value and returns a PDF color string
`glyph`	`show`	`false`	Whether to mark glyphs
	`color`	`1 0 0 RG`	PDF stroking color operator
	`width`	`0.1`	Line width in bp units
	`baseline`	`true`	Whether to mark the baseline
`onlyglyphs`	—	—	Shortcut to disable all markers except glyphs
`opacity`	—	(empty)	PDF graphics state operator for transparency

Notes:

- The `kern` key uses both stroke and fill colors, unlike other keys which only

use stroking color.

- The `onlyglyphs` key is a boolean flag (no value needed) that sets all `show` keys to `false` except `glyph/show`, which is set to `true`.
- The `opacity` key applies to all node types. For fine-tuned opacity control per node type, the `color` keys can be (ab)used to include graphics state operators.

The default `colorfunc` for penalties is:

```
function(p)
  local color = "1 g"
  if p < 10000 then
    color = string.format("%g g", 1 - math.floor(p / 10000))
  end
  return color
end
```

An example of the usage of the keys is

```
\RequirePackage{pdfmanagement}
\UseName{pdfmanagement_add:nnn}
  {Page/Resources/ExtGState}{lvd}
  {<</ca 0.5/CA 0.5>>}
\documentclass[border=5pt]{standalone}
\usepackage{lua-visual-debug}
\usepackage{unicode-math}
\lvdset{
  glyph = {color={0 0 1 RG} width=0.12}
  onlyglyphs
  opacity = {/lvd gs}
}
\begin{document}
This is an example $\intop_a^b f(x) dx$
\end{document}
```

which yields

You can also access and assign the parameters from lua with the `params` table. For example

```
local params = require('lua-visual-debug').params
params.hlist.color = "1 0 0 RG"
```

7 Copying