

# Package `highlightlatex` manual

Vincent Kuhlmann  
vincent.kuhlmann@hotmail.com

March 20, 2021

## Abstract

This package provides colored syntax highlighting for L<sup>A</sup>T<sub>E</sub>X source code, without aid from outside L<sup>A</sup>T<sub>E</sub>X. This is in response to the general trend that people often fall back to verbatim for displaying code. This package aims to make good looking L<sup>A</sup>T<sub>E</sub>X source code feasible for all users. For this, it builds further on the generic ‘listings’ package. An example output is shown in Figure 1.

## Repository

<https://github.com/vkuhlmann/highlight-latex>

---

And look at this beautiful code

```
% Here is some code
\setcounter{secnumdepth}{1}
\begin{document}
  \section{My section (and Hi!)}

  \unknowncommand\knowncommand
  \inAccA\inAccB\inAccC
  \section  $\sqrt{2}$  ← cool!

  Insert literal tildes like ~. Hi!
\end{document}
```

with some text after it.

Let’s also show square brackets:

```
% File: document.tex
\documentclass[a4paper]{article}
\usepackage{amsmath}

\begin{document}
  Insert math like  $\sqrt{2}$ $.
  \inAccA\inAccB\inAccC

\end{document}
```

The first line was of the form `\documentclass[]{}.` Very interesting. Note the `$` are only green because we defined it as a keyword.

Go to <https://github.com/vkuhlmann/highlight-latex>

Figure 1: Output of ‘demo.tex’.

---

# Contents

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Inline style . . . . .	3
1.2	Block style . . . . .	3
<b>2</b>	<b>Macro <code>\updatehighlight</code></b>	<b>3</b>
2.1	Adding a command to a highlighting rule . . . . .	3
2.2	Custom highlighting rules . . . . .	4
2.2.1	Example . . . . .	4
2.2.2	Specification . . . . .	4
<b>3</b>	<b>Global settings</b>	<b>5</b>
<b>4</b>	<b>Fragile breaking situations (like beamer frames)</b>	<b>5</b>
<b>5</b>	<b>Adding extra space</b>	<b>6</b>
<b>6</b>	<b>License</b>	<b>6</b>
<b>7</b>	<b>Credits</b>	<b>6</b>
<b>8</b>	<b>Implementation</b>	<b>7</b>
8.1	Package options . . . . .	8
8.2	Initialization . . . . .	8
8.3	Options . . . . .	9
8.4	Block and inline markup . . . . .	10
8.5	Block saving . . . . .	10
8.6	Macro <code>\updatehighlight</code> . . . . .	11
8.6.1	Key ‘name’ . . . . .	12
8.6.2	Key ‘classoffset’ . . . . .	12
8.6.3	Key ‘add’ . . . . .	12
8.6.4	Key ‘remove’ . . . . .	13
8.6.5	Key ‘style’ . . . . .	14
8.6.6	Key ‘color’ . . . . .	14
8.6.7	Key ‘clear’ . . . . .	14

# 1 Getting started

After having added the package, you can add LaTeX in two ways.

## 1.1 Inline style

EXAMPLE

```
Your file begins with a line of the form \hll|\documentclass[]{}|. The
square brackets ...
```

The first non-space character following `\hll` delimits the argument to this command.

## 1.2 Block style

EXAMPLE

```
Your basic document now looks like
\begin{highlightblock}[gobble=2]
  \documentclass[a4paper]{article}
  \begin{document}
    Hello world!
  \end{document}
\end{highlightblock}
```

To prevent indentation of our `highlightblock` (here one tab) to be shown as part of the code, the `gobble` parameter strips them off. Play around with it until everything looks right. I recommend to set this value globally using `\def\defaultgobble{2}`. You can still override it on a per-block basis, if necessary.

There are situations where width of the block could run out of the page. For example, when using beamer and storing a block as described in the section ‘Fragile breaking situations’, the normal full-width of a slide is assumed. If you use multiple columns, set the `linewidth` on the `highlightblock`. This can be a fraction of the total slide width available, `0.6\textwidth` is 60% of the width, or an absolute value, like `10em`, which seems to equal 20 characters.

There are more keys you can provide. Check the `listings` package documentation<sup>1</sup> for options available to the `lstlisting`-environment and `lstset` command.

# 2 Macro `\updatehighlight`

## 2.1 Adding a command to a highlighting rule

By default, only some LaTeX commands will be highlighted in blue. If there are others you need, like `\tableofcontents` and `\figref`, update the highlighting rules:

USE

```
\updatehighlight{
  name = default,
  add = {
    \tableofcontents, \figref
  }
}
```

The change will only affect code after it. I recommend issuing `updatehighlight` in your preamble (before the `\begin{document}`). In some situations you might want to change things mid-document. That’s possible too.

---

<sup>1</sup> <https://www.ctan.org/pkg/listings>

## 2.2 Custom highlighting rules

### 2.2.1 Example

As shown in `demo.tex`, you can put any command or keyword you want to highlight in a different color. You do this with

EXAMPLE

```
\updatehighlight{
  % The name allows you to modify the style later.
  name = spotlight,
  color = orange,
  add = {
    \tableofcontents
  }
}
```

You can use the `xcolor` syntax for describing colors as well. If you find the orange too bright, you can replace it with `orange!90!black`: 90% orange, remaining is black. For more information on color definitions and name, refer to [LaTeX/Colors on Wikibooks](https://en.wikibooks.org/wiki/LaTeX/Colors)<sup>2</sup>.

### 2.2.2 Specification

The argument to `\updatehighlight` is a key-value list. Keys are processed sequentially. For example, use `color` before `add` rather than after it, and a key can appear multiple times. Each one will be processed. You can merge any two `\updatehighlight` in one. No need to close and reopen `\updatehighlight` for each highlighting rule.

You might be tempted to add a blank line for clarity; that means a new paragraph to LaTeX, don't do it. Instead, just put a line with only a `%` sign. Spacing within the argument is often irrelevant. If you need a comma in the value, surround your value with braces.

#### **name**

Creates or modifies a named rule. This key is optional.

The default keys are `default`, which includes a bunch of basic commands, and has by default a dark blue color, and `structure`, which consists of `\begin` and `\end` and prints them in light blue.

#### **classoffset**

Sets the `listings` classoffset manually. Try to avoid this. Use **name** to refer to existing rules instead.

#### **add**

Adds a command (`\mycommand`) or keyword (`Hi!`) to the current rule. The value can contain multiple values by opening braces, and comma separating values within them.

#### **remove**

Removes a commands or keywords from the current rule. The value can contain multiple values by opening braces, and comma separating values within them.

---

<sup>2</sup> <https://en.wikibooks.org/wiki/LaTeX/Colors>

## clear

Removes all commands and keywords from the current rule. Use without value, for example

### EXAMPLE

```
\updatehighlight{
  name = default,
  clear
}
```

## color

Specifies a color for the rule. Equivalent to specifying `style` instead, with value `\color{value}` where `value` is the value for the `color` key. So `color=red` and `style=\color{red}` are equivalent.

## style

Specifies a style for the rule. A rule can have only one style. If you specify a style after `add` or `remove`, this starts a new (unnamed) rule. In practice, the only style which will probably work for you is just a color. For that, using the ‘color’ key is just a bit easier and neater. But hey, you have the option to set whatever style you want. :)

## 3 Global settings

There are some global parameters involved in the appearance:

### OVERVIEW

```
\colorlet{curlyBrackets}{red!50!blue}
\colorlet{squareBrackets}{blue!50!white}
\colorlet{codeBackground}{gray!10!white}
\colorlet{comment}{green!40!black}
\def\defaultgobble{0}
```

Each line can be set independent of each other, and each shows its default value.

There are package options you can use as well:

**frame** (default: `lines`)

Specifies the frame you want around code. My favorites are `lines` and `none`. Check the listings package documentation <sup>3</sup> for all possibilities.

**noframe** (use without value)

Equivalent to `frame=none`.

**styleanywhere** (use without value)

Overrides the default behavior that `style` starts a new style after commands like `add` and `remove`.

## 4 Fragile breaking situations (like beamer frames)

When passing command arguments around, or storing environment content, LaTeX interprets all characters. This includes seeing `\maketitle` in `\hll|\maketitle|` as a real command. To prevent this behavior, everything from `\verb`, to the `verbatim`-environment, to the `listings` package (which this package depends on) temporarily changes the interpretation of characters that are still to be read. The backslash before `\maketitle` in `\hll|\maketitle|` will be read as ‘just text’ (a *letter* technically).

<sup>3</sup> <https://www.ctan.org/pkg/listings>

When content has already been interpreted, like the `frame`-environment in `beamer` does, this trick can't be done anymore. Instead, you either need to *escape* code, or *pre-process* the code outside a fragile breaking situation.

Escaping is done by preceding the special character with a backslash. For example, `\hll|\documentclass[]{}|` becomes `\hll|\documentclass[]\{\}|`.

For large code blocks, this is undesirable. Therefore, the package provides for a companion to the `highlightblock`-environment: surround it with a `saveblock`-environment which takes a single argument: a name to assign it. We use it to refer to it later. For example:

EXAMPLE

```
\begin{saveblock}{basicfigure}
\begin{highlightblock}[linewidth=0.6\textwidth]
\begin{figure}
\includegraphics
[width=0.9\linewidth]
{myPlot.pdf}

\caption{My plot}
\label{fig:myplot}
\end{figure}
\end{highlightblock}
\end{saveblock}
```

Do this outside a fragile breaking situation. (For the `frame`-environment example, that means just before the frame for example.) Then, where you want to use it, use `\useblock{basicfigure}`. There is also a variant `\consumeblock{basicfigure}`. If you save many blocks, these will all remain loaded in memory till your PDF has fully generated. The `\consumeblock` works like `\useblock`, except the saved block is deleted from memory after its use. Note this can also result in unexpected behavior, for example animations in a `beamer` frame might need the code line to be executed multiple times. Use `\useblock` when you can't make the guarantee the last use of a block.

There is a separate demo for fragile breaking situations. You can find it at `deamerdemo/deamerdemo.tex`.

## 5 Adding extra space

By default, `highlight-latex` follows an approach where it minimizes spacing. This gives you full control over how tight or spacious your document looks. Just use commands like `\medskip` to add extra spacing. The package doesn't currently include an option to do it everywhere automatically.

## 6 License

The package is available under MIT License. See `LICENSE.txt`.

## 7 Credits

Thanks for minor fixes:

gemmaro

For any bug, feature request, unclarity, or whatever else related to this package, you're welcome to open an issue or pull-request. Issues can be opened on

---

URL      <https://github.com/vkuhlmann/highlight-latex/issues>

---

Thanks for contributing!

## 8 Implementation

```
1 \NeedsTeXFormat{LaTeX2e}[1994/06/01]
2 \ProvidesPackage{highlightlatex}[2021/03/17 highlightlatex v1.0.0 + Dev]
3
4 % Repository: https://github.com/vkuhlmann/highlight-latex
5 % Version: v1.0.0 + Dev
6
7 % Copyright (c) 2021 Vincent Kuhlmann
8 %
9 % Permission is hereby granted, free of charge, to any person
10 % obtaining a copy of this software and associated documentation
11 % files (the "Software"), to deal in the Software without
12 % restriction, including without limitation the rights to use,
13 % copy, modify, merge, publish, distribute, sublicense, and/or sell
14 % copies of the Software, and to permit persons to whom the
15 % Software is furnished to do so, subject to the following
16 % conditions:
17 %
18 % The above copyright notice and this permission notice shall be
19 % included in all copies or substantial portions of the Software.
20 %
21 % THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
22 % EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
23 % OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
24 % NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
25 % HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
26 % WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
27 % FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
28 % OTHER DEALINGS IN THE SOFTWARE.
29
30 \RequirePackage{listings}
31 \RequirePackage{xcolor}
32
33 \RequirePackage{xkeyval}
34 \RequirePackage{etoolbox}
35
36 %% Snippet source:
37 %% https://tex.stackexchange.com/questions/406015/defining-macro-gsetlength-as-global-setlength-reliable
38 \gdef\gsetlength#1#2{%
39   \begingroup
40     \setlength\skip@{#2}%
41     \global#1=\skip@
42   \endgroup
43 }
44
45
46 \def\@highlight@frame{none}
47 \newlength{\hll@offset@fixskip}
48 \setlength{\hll@offset@fixskip}{0pt}
49
50 \newif\ifhllStyleAnywhere
51 \newcommand\defaultgobble{0}
52 \newif\ifhllDebugFrame
53 %\hllDebugFrame true
54 \newlength\hllBeforeSkip
55 \setlength\hllBeforeSkip{2pt}
56 \newlength\hllAfterSkip
57 \setlength\hllAfterSkip{0pt}
```

## 8.1 Package options

```

59 \DeclareOptionX{frame}{
60   \setframe{#1}
61 }
62
63 \DeclareOptionX{noframe}{
64   \setframe{none}
65 }
66
67 \def\setframe#1{%
68   \def\frameval{#1}%
69   \ifx\relax#1\relax
70     \def\frameval{lines}%
71   \fi
72   \def\linesvalue{lines}%
73   %
74   \xdef\@highlight@frame{\frameval}%
75   %
76   \ifx\frameval\linesvalue
77     \gsetlength{\hll@offset@fixskip}{6pt}%
78   \else
79     \gsetlength{\hll@offset@fixskip}{0pt}%
80   \fi
81 }
82
83 \DeclareOptionX{styleanywhere}{
84   \hllStyleAnywheretrue
85 }
86
87 \DeclareOptionX*{\PackageWarning{highlightlatex}{‘\CurrentOption’ ignored}}
88 \ExecuteOptionsX{frame=lines}
89
90 \ProcessOptionsX\relax
91
```

## 8.2 Initialization

```

92 \colorlet{curlyBrackets}{red!50!blue}
93 \colorlet{squareBrackets}{blue!50!white}
94 \colorlet{codeBackground}{gray!10!white}
95 \colorlet{comment}{green!40!black}
96
97 \colorlet{accDefault}{blue!90!black}
98 \definecolor{accStructure}{RGB}{0,149,255}
99
100 \colorlet{accentB}{green!60!black}
101
102 \catcode‘\{=11
103 \let\myopenbrace={
104 \catcode‘\{=1
105
106 \catcode‘\}=11
107 \let\myclosebrace=}
108 \catcode‘\}=2
109
110 \lstdefinlanguage{ColoredLaTeX}
111 [LaTeX]{TeX}{

```



```

112 backgroundcolor=\color{codeBackground},
113 keywordstyle=\bfseries\color{accDefault},
114 basicstyle=\footnotesize\ttfamily,
115 commentstyle=\color{comment},
116 numbers=none,
117 tabsize=2,
118 mathescape=false,
119 alsoletter={$_!|?}$},
120 %
121 classoffset=0,
122 keywordstyle=\color{accDefault},
123 texcsstyle=\color{accDefault},
124 moretexcs={subsection,thesubsection,thesection,theoremstyle},
125 deletetexcs={begin,end},
126 %
127 classoffset=1,
128 keywordstyle=\color{accStructure},
129 texcsstyle=\color{accStructure},
130 moretexcs={begin,end},
131 %
132 classoffset=2,
133 escapeinside=~~,
134 %
135 classoffset=0,
136 %literate={textA}{textB}{3}%
137 %https://tex.stackexchange.com/questions/172945/coloring-in-listing
138 literate={\}{\textcolor{curlyBrackets}{\myopenbrace}}{1}
139      {\}{\textcolor{curlyBrackets}{\myclosebrace}}{1}
140      {\}{\textcolor{squareBrackets}{[]}}{1}
141      {\}{\textcolor{squareBrackets}{[]}}{1}%
142 %      {\}{\textcolor{delimiterColor}{()}}{1}
143 %      {\}{\textcolor{delimiterColor}{()}}{1}%
144 %texcsstyle=\color{blue}
145 }
146 \lstset{language=ColoredLaTeX}
147
148 \def\hll@assignlabel#1#2{%
149   \expandafter\gdef\csname hll@labeltoclassoffset@#1\endcsname{#2}%
150 }
151
152 \hll@assignlabel{default}{0}
153 \hll@assignlabel{structure}{1}
154

```

### 8.3 Options

```

155
156 \newcommand{\hllDelimDollar}[1][\color{green!40!black}]{%
157   \lstset{
158     classoffset=2,
159     moredelim=**[s][#1]{$}{$},
160     classoffset=0
161   }%
162 }
163
164 \newcommand{\hllDelimParen}[1][\color{green!40!black}]{%
165   \lstset{
166     classoffset=2,
167     moredelim=**[s][#1]{\{()\}},

```

```

168     classoffset=0
169 }%
170 }
171

```

## 8.4 Block and inline markup

```

172
173
174 \newcommand{\hllPrepareBlock}{%
175   \lstset{framesep=0.1cm,framerule=0.6pt}%
176   \def\setHighlightFrame##1{%
177     \lstset{frame=##1}%
178   }%
179   \expandafter\setHighlightFrame\expandafter{\@highlight@frame}%
180   \lstset{belowskip=0pt,aboveskip=0pt,belowcaptionskip=0pt,gobble=\defaultgobble}%
181   \setlength\parskip{0pt}%
182   \setlength\parindent{0pt}%
183 }
184
185 \lstnewenvironment{highlightblock}[1] []
186 {%
187   \vspace{\hllBeforeSkip}%
188   \begingroup
189   \hllPrepareBlock\lstset{#1}%
190 }{%
191   \nointerlineskip
192   \endgroup
193   \vspace{\hllAfterSkip}%
194 }
195
196 \lstnewenvironment{hllblock}[1] []
197 {%
198   \vspace{\hllBeforeSkip}%
199   \begingroup
200   \hllPrepareBlock\lstset{#1}%
201 }{%
202   \endgroup
203   \vspace{\hllAfterSkip}%
204 }
205
206 \def\hll{\lstinline}
207

```

## 8.5 Block saving

```

208
209
210 \newenvironment{saveblock}[1]{%
211   \expandafter\ifx\csname @codebox@#1\endcsname\relax
212     \expandafter\newbox\csname @codebox@#1\endcsname\relax
213   \fi
214   %
215   %% The trick from latex/base/latex.ltx
216   \edef\@bracehack{%
217     \endgroup
218     \expandafter\setbox\csname @codebox@#1\endcsname\vbox{%
219       \begingroup\aftergroup}%
220   \def\noexpand\@currentvir{\@currentvir}%
221   \def\noexpand\@currentvline{\on@line}}%

```

```

222 \@bracehack
223 \begingroup\vspace{\hll@offset@fixskip}
224 \ignorespaces%
225 }{%
226 \endgroup
227 }
228
229 \newcommand\useblock[1]{%
230 \ifhllDebugFrame
231 \par
232 {%
233 \setlength{\fboxsep}{0pt}%
234 \fcolorbox{orange}{red}{\expandafter\usebox\csname @codebox@#1\endcsname}%
235 }%
236 \par
237 \else
238 \par\expandafter\usebox\csname @codebox@#1\endcsname\par
239 \fi
240 }
241
242 \newcommand\consumeblock[1]{%
243 \ifhllDebugFrame
244 \par
245 {%
246 \setlength{\fboxsep}{0pt}%
247 \setlength{\fboxrule}{1pt}%
248 \fcolorbox{orange}{red}{\leavevmode\expandafter\box\csname @codebox@#1\endcsname\relax}%
249 }
250 \par
251 \expandafter\let\csname @codebox@#1\endcsname\relax
252 \else
253 \par\leavevmode\expandafter\box\csname @codebox@#1\endcsname\relax\par
254 \expandafter\let\csname @codebox@#1\endcsname\relax
255 \fi
256 }
257

```

## 8.6 Macro \updatehighlight

```

258 \def\hll@update@label{}
259
260 \newcounter{hll@update@state}
261
262 \newcounter{hll@update@nextclassoffset}
263 \setcounter{hll@update@nextclassoffset}{5}
264
265 \newcounter{hll@update@classoffset}
266
267 \def\hll@update@fixclassoffset{%
268 \ifnum\value{hll@update@state}<1\relax
269 \setcounter{hll@update@classoffset}{\thehll@update@nextclassoffset}%
270 \stepcounter{hll@update@nextclassoffset}%
271 \setcounter{hll@update@state}{1}%
272 \fi
273 }
274
275 \newcommand{\updatehighlight}[1]{%
276 \setcounter{hll@update@state}{-1}%
277 \gdef\hll@update@label{}%

```

```

278 \setkeys{updatehighlight}{#1}%
279 %
280 \lstset{classoffset=0}%
281 }
282

```

### 8.6.1 Key ‘name’

```

283 \define@key{updatehighlight}{name}{%
284 \hll@update@setlabel{#1}%
285 }
286
287 \def\hll@update@setlabel#1{
288 \gdef\hll@update@label{#1}%
289 \expandafter\def\expandafter\@capturedlabel\expandafter{\csname hll@labeltoclassoffset@#1\endcsname}%
290 \expandafter\unless\expandafter\ifx\@capturedlabel\relax\relax
291 % Label exists
292 \edef\@val{\@capturedlabel}%
293 \setcounter{hll@update@classoffset}{\@val}%
294 \setcounter{hll@update@state}{1}%
295 \else
296 % Allocate new label
297 \edef\@val{\the\hll@update@nextclassoffset}%
298 \setcounter{hll@update@classoffset}{\@val}%
299 \stepcounter{hll@update@nextclassoffset}%
300 \setcounter{hll@update@state}{1}%
301 %
302 \expandafter\xdef\@capturedlabel{\@val}%
303 \fi
304 }
305

```

### 8.6.2 Key ‘classoffset’

```

2 % macro:->end{highlightblock}
3 \define@key{updatehighlight}{classoffset}{%
4 \setcounter{hll@update@classoffset}{#1}%
5 \setcounter{hll@update@state}{1}%
6 }
7 % tadaaa
8 %

```

---

```

aaa

```

### 8.6.3 Key ‘add’

```

306 \define@key{updatehighlight}{add}{%
307 \hll@update@add{#1}%
308 }
309
310 \def\hll@update@add#1{%
311 \hll@update@fixclassoffset
312 \def\@inv##1{\forcsvlist{\hll@update@additem{##1}}{#1}}%
313 \expandafter\@inv\expandafter{\the\c@hll@update@classoffset}
314 %
315 \setcounter{hll@update@state}{2}%
316 }
317
318 \def\hll@update@additem#1#2{%
319 \if\noexpand#2\relax
320 \edef\keywordpure{\expandafter\gobble\string#2}%
321 \def\callwithpure##1{\hll@update@addmacro{#1}{##1}}%

```

```

322 \expandafter\callwithpure\expandafter{\keywordpure}%
323 \else
324 \hll@update@addkeyword{#1}{#2}%
325 \fi
326 }
327
328 \def\hll@update@addmacro#1#2{%
329 \lstset{
330 classoffset=0,deletetexcs={#2},
331 classoffset=1,deletetexcs={#2},
332 classoffset=#1,moretexcs={#2},
333 classoffset=0
334 }%
335 }
336
337 \def\hll@update@addkeyword#1#2{%
338 \lstset{
339 classoffset=0,deletekeywords={#2},
340 classoffset=1,deletekeywords={#2},
341 classoffset={#1},otherkeywords={#2},morekeywords={#2},
342 classoffset=0
343 }%
344 }
345

```

#### 8.6.4 Key ‘remove’

```

346 \define@key{updatehighlight}{remove}{%
347 \hll@update@remove{#1}%
348 }
349
350 \def\hll@update@remove#1{%
351 \hll@update@fixclassoffset
352 \def\@inv##1{\forcsvlist{\hll@update@removeitem{##1}}{#1}}%
353 \expandafter\@inv\expandafter{\the\c@hll@update@classoffset}
354 %
355 \setcounter{hll@update@state}{2}%
356 }
357
358 \def\hll@update@removeitem#1#2{%
359 \if\noexpand#2\relax
360 \edef\keywordpure{\expandafter\@gobble\string#2}%
361 \def\callwithpure##1{\hll@update@removemacro{#1}{##1}}%
362 \expandafter\callwithpure\expandafter{\keywordpure}%
363 \else
364 \hll@update@removekeyword{#1}{#2}%
365 \fi
366 }
367
368 \def\hll@update@removemacro#1#2{%
369 \lstset{
370 classoffset={#1},
371 deletetexcs={#2},
372 classoffset=0
373 }%
374 }
375
376 \def\hll@update@removekeyword#1#2{%
377 \lstset{

```

```

378     classoffset={#1},
379     deletekeywords={#2},
380     classoffset=0
381 }%
382 }
383

```

### 8.6.5 Key ‘style’

```

384 \define@key{updatehighlight}{style}{%
385   \hll@update@fixclassoffset
386   \unless\ifhllStyleAnywhere
387     \ifnum\value{hll@update@state}>1\relax
388       \setcounter{hll@update@state}{0}%
389       \hll@update@fixclassoffset
390     \fi
391   \fi
392   %
393   \expandafter\hll@update@setstyle\expandafter{\the\c@hll@update@classoffset}{#1}%
394 }
395
396 \def\hll@update@setstyle#1#2{%
397   \lstset{%
398     classoffset={#1},
399     keywordstyle={#2},
400     texcsstyle={*#2},
401   }%
402 }
403

```

### 8.6.6 Key ‘color’

```

404 \define@key{updatehighlight}{color}{%
405   \hll@update@fixclassoffset
406   \unless\ifhllStyleAnywhere
407     \ifnum\value{hll@update@state}>1\relax
408       \setcounter{hll@update@state}{0}%
409       \hll@update@fixclassoffset
410     \fi
411   \fi
412   %
413   \expandafter\hll@update@setcolor\expandafter{\the\c@hll@update@classoffset}{#1}%
414 }
415
416 \def\hll@update@setcolor#1#2{%
417   \lstset{%
418     classoffset={#1},
419     keywordstyle={\color{#2}},
420     texcsstyle={*\color{#2}},
421   }%
422 }
423

```

### 8.6.7 Key ‘clear’

```

424 \define@key{updatehighlight}{clear}[]{}%
425   \hll@update@fixclassoffset
426   \expandafter\hll@update@clear\expandafter{\the\c@hll@update@classoffset}%
427 }
428
429 \def\hll@update@clear#1{%

```

```
430 \lstset{
431   classoffset={#1},
432   texcs={},
433   keywords={},
434   classoffset=0
435 }%
436 }
```