

# THE OSGLISTINGS PACKAGE

VO.1.0      2026-08-28

A [minted](#)-based front end for code listings

Matthias WERNER<sup>1</sup>

<https://github.com/werner-matthias/osglecture>

The **OSGLISTINGS** package provides a frontend for [minted](#) that makes colors and border styles easily configurable and supports highlights, external editors, and code sources (gists).

In addition, the source code is tagged as Code if `\DocumentMetadata{tagging=on}` is set.

The package is part of the [osglecture](#) bundle, but is largely independent of the other packages.

## Table of Contents

|                             |          |                          |          |
|-----------------------------|----------|--------------------------|----------|
| <b>1 Usage</b>              | <b>1</b> | <b>1.6 Accessibility</b> | <b>6</b> |
| 1.1 Box Styles              | 2        | 1.7 Tagging              | 7        |
| 1.2 Color Palette           | 2        |                          |          |
| 1.3 Links to Online Editors | 3        | <b>2 Implementation</b>  | <b>7</b> |
| 1.4 Remote Sources          | 3        |                          |          |
| 1.5 Character-level Marks   | 4        | <b>Index</b>             | <b>8</b> |

## 1 Usage

The package is loaded in the usual way using `\usepackage` and, in turn, loads (among other things) [minted](#).

[osglistings](#) knows four package options: `cache`, `cachedir`, `frozencache`, and `cacheignoresfilecontents`. They are forwarded unchanged to [minted](#)'s own, identically named load-time options, controlling its highlighting cache. Without them, [minted](#) behaves as with its own defaults. Other [minted](#) package options currently cannot be set through [osglistings](#).

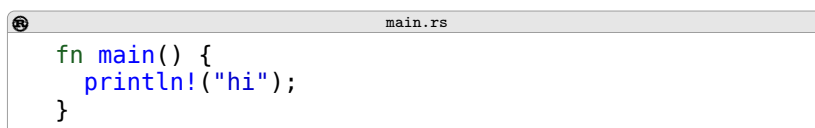
---

1. [matthias.werner@informatik.tu-chemnitz.de](mailto:matthias.werner@informatik.tu-chemnitz.de)

```
1 \usepackage[cachedir=build/minted-cache]{osglistings}
```

Source that lives directly in the document is typeset with the `osglisting` environment; source from an external file with `\osglistinginput`. Both accept the same keys, plus any option `minted` itself understands (`firstline`, `breaklines`, ...), forwarded unchanged.

```
1 \documentclass{article}
2 \usepackage{osglistings}
3 \begin{document}
4   \begin{osglisting}[style=editor, title=main.rs]{rust}
5     fn main() {
6       println!("hi");
7     }
8   \end{osglisting}
9 \end{document}
```



## 1.1 Box Styles

The `style` key chooses between three looks: `plain` (a plain frame), `paper` („printer paper“ with a torn edge, no title bar), and `editor` (an „editor window“ with a title bar, a language badge, and, when `link` is set, an edit-link icon). The `paper` style does not support breaking across a page.

## 1.2 Color Palette

Syntax colors come from a named palette, activated with `\UseOsgListingsPalette`, or for a single call via the `palette` key. A custom palette is declared with `\DeclareOsgListingsPalette`; slots left unspecified inherit their value from the built-in default palette `osglistings-default`.

```

1 \DeclareOsgListingsPalette{my-theme}{
2   keyword = {model=HTML, value=8A2BE2},
3   string  = {alias=keyword},
4 }
5 \UseOsgListingsPalette{my-theme}

```

Each slot is either a *resource* (model+value, in any model `xcolor` supports) or an *alias* (alias=*other slot*), resolved at most two levels deep).

### 1.3 Links to Online Editors

The link key attaches a URL; link-display controls where it shows up: titlebar (an icon in the title bar, editor style only), label (clickable text below the box), qrcode (a QR code below the box), or none (no visible indicator).

### 1.4 Remote Sources

The third argument of `\osglistinginput` accepts, besides a local path, a gist: or github: locator as well, e.g. `gist:<owner>/<gist-id>/<filename>` or `github:<owner>/<repo>@<ref>/<path>`. The source is fetched on first compile and cached under `\jobname.osglistings-cache/`; cache controls reuse: `cache=true` (the default) uses an existing cache entry, `cache=refresh` always re-fetches and updates it.

```

1 \osglistinginput[style=editor, title={fizzbuzz.rs}]
2 {rust}{gist:werner-matthias/3af1e0.../fizzbuzz.rs}

```

When link is not set, the package automatically derives an editor link from the language registry: from editor-url-gist, if declared for the language and the source is a gist (template typically using `{GIST_ID}` instead of `{CODE}`), else from editor-url. The `{CODE}` placeholder is replaced with the URL-encoded source, `{GIST_ID}` with the gist id.

An entry, once cached, is not automatically validated against its source on compile – `remote-check>manual` (the default) checks nothing; `remote-check=compile` fires exactly one request per *distinct* repository/gist at the end of the document (not per file) and warns about stale entries; `remote-refresh=true` additionally refreshes stale entries immediately. Outside L<sup>A</sup>T<sub>E</sub>X, the command-line script `osglistings-check-updates.lua` runs the same check for one or more cache directories.

```
1 texlua osglistings-check-updates.lua --refresh myfile.osglistings-cache
```

### Attention, Security Risk!

A `gist:` or `github:` locator requires compiling with `--shell-escape` and internally invokes `curl` with the T<sub>E</sub>X process's own permissions. Only use these locators with documents (and locators) you trust – a maliciously chosen locator could otherwise trigger arbitrary network requests under your identity.

## 1.5 Character-level Marks

`\osglistingsmark{name}` drops an invisible, named `tikz` node exactly where it appears in the source; a later `tikzpicture` can then draw to or from it. For inline source (the `osglisting` environment) `escapeinside` must be set (e.g. `escapeinside=$$`) for `\osglistingsmark{name}` to be recognized as L<sup>A</sup>T<sub>E</sub>X code at all, rather than verbatim text.

```
1 \documentclass{article}
2 \usepackage{tikz}
3 \usepackage{osglistings}
4 \begin{document}
5   \begin{osglisting}[style=plain, escapeinside=$$]{rust}
6     let code = ExitCode::from(42\$\osglistingsmark{here}\$);
7   \end{osglisting}
8   \begin{tikzpicture}[remember picture, overlay]
9     \draw[->, red, thick] (osgm-here) ++(0,1) node[above]{Exit-Code}
10      -- (osgm-here);
11 \end{tikzpicture}
12 \end{document}
```

Exit-Code



```
let code = ExitCode::from(42);
```

For external sources, the marks key of `\osglistinginput` does the same job without needing

`escapeinside` set manually – when `marks` is non-empty, the package switches on `escapeinside` itself (delimiter `§`, overridable via `marks-escape`). Each comma-separated entry addresses a position either exactly by `<name>` at `<line>:<col>` (`<col>` counts in *characters*, not bytes) or via a literal text search using `<name>` after "`<text>`" or `<name>` before "`<text>`", optionally followed by `occurrence=<n>` (default 1).

```

1 \osglistinginput[
2   marks = {
3     exitc after "ExitCode",
4     semi before ";" occurrence=2,
5   }
6 ]{rust}{src/example.rs}

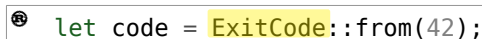
```

`around` behaves like `after/before`, but instead of marking a single spot it brackets the found text with a node pair `<name>-begin/<name>-end`. `\osglistingshighlight` then connects such a pair into a highlight box – thanks to the character height/depth captured by `\osglistingsmark`, with no manual `xshift/yshift` corrections needed. For inline source the same effect is available without `marks` too, by simply placing `\osglistingsmark` twice (named `<name>-begin/<name>-end`) directly in the source.

```

1 \documentclass{article}
2 \usepackage{tikz}
3 \usepackage{osglistings}
4 \begin{document}
5   \begin{osglisting}[style=plain, escapeinside=§§]{rust}
6     let code = §\osglistingsmark{kw-begin}§ExitCode§\osglistingsmark{kw-end}§::from
7       (42);
8   \end{osglisting}
9   \osglistingshighlight{kw}
10 \end{document}

```



```
let code = ExitCode::from(42);
```

For external sources, around in the marks entry is enough instead, followed by `\osglistingshighlight`.

```
1 \osglistinginput[
2   marks = { exitc around "ExitCode" }
3 ]{rust}{src/example.rs}
4 \osglistingshighlight{exitc}
```

marks (like any use of `\osglistingsmark`) requires Lua<sub>A</sub>T<sub>E</sub>X.

## 1.6 Accessibility

When an around entry also carries `desc="⟨text⟩"`, the substitution opens a tagged Span structure instead of a bare point pair, with that text as its `alt` attribute (requires `\DocumentMetadata{tagging=on}`; without it, a plain point pair as usual) – assistive technology then reads `desc="⟨text⟩"` instead of the marked-up source itself. For inline source, `\osglistingsmarkspanbegin/\osglistingsmarkspanend` do the same by hand. `\osglistingshighlight` itself wraps its overlay as an Artifact automatically; assistive technology skips it, because the real information already hangs off the source via `desc="⟨text⟩"`, not off the arrow or box around it.

```
1 \osglistinginput[
2   marks = {
3     exitc around "ExitCode" desc="the process exit status",
4   }
5 ]{rust}{src/example.rs}
6 \osglistingshighlight{exitc}
```

### Only Tagged Spans Carry Text

The rest of the source does land inside a tagged Code structure, but (as currently implemented) without screen-reader-readable text – for a marked span, desc is therefore usually not an addition but the only information actually read aloud.

### Second Compile Run Needed

remember `picture` only resolves correctly, like any cross-page `tikz` overlay, after a second compile run – before the first run (or after changing the marked positions) arrows/markers still point at their previous position or not at all.

## 1.7 Tagging

When L<sup>A</sup>T<sub>E</sub>X is started with `\DocumentMetadata{tagging=on}`, the source is tagged as structure Code (structure name `osglistings/code`, remappable with `\NewStructureName/\AssignStructureRole`). Tagging is switched off per call with `tag=false`.

## 2 Implementation

To view the implementation description, please uncomment `osglistings.dtx` in `osgdoc.dtx`.

# Index

|                                  |         |                                                          |       |
|----------------------------------|---------|----------------------------------------------------------|-------|
| <b>A</b>                         |         | <b>O</b>                                                 |       |
| \AssignStructureRole .....       | 7       | \osglistinginput .....                                   | 2 ff. |
| <b>D</b>                         |         | \osglistingshighlight .....                              | 5 f.  |
| \DeclareOsgListingsPalette ..... | 2       | \osglistingsmark .....                                   | 4 ff. |
| \DocumentMetadata .....          | 1, 6 f. | \osglistingsmarkspanbegin .....                          | 6     |
| <b>J</b>                         |         | \osglistingsmarkspanend .....                            | 6     |
| \jobname .....                   | 3       | <b>U</b>                                                 |       |
| <b>N</b>                         |         | \UseOsgListingsPalette .....                             | 2     |
| \NewStructureName .....          | 7       | \usepackage[ <i>&lt;options&gt;</i> ]{osglistings} ..... | 1     |