

THE SEMCAT PACKAGE

VO.3.0 2026-08-31

Semantic category markup

Matthias WERNER¹

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

`osglecture``osglecture-modes\DocumentMetadata{tagging=on}`

Table of Contents

1 Introduction	1	2.4 QR-code annotations	3
2 Usage	1	2.5 Mode and tagging integration	4
2.1 Categories	1	3 Implementation	4
2.2 Marking	3	Index	5
2.3 Explanation box	3		

1 Introduction

2 Usage

2.1 Categories

`\SemCatDefine{<label>}[<options>]`

`\SemCatDefine*{<label>}[<options>]`

`title = {<description>}`

`slot = {<index>}`

`color = {<color>}`

1. matthias.werner@informatik.tu-chemnitz.de

```
\SemCatDefine{A}[title={Historical background}, color=teal]
\SemCatDefine*{B}[title={Core material}, slot=primary]
\SemCatDefine{C}[title={Comparison}, index=7]
```

The star form makes the category the new default. Without a color key, the color is assigned round-robin from the dedicated `semcat-categories` color series while `osgstyler` is loaded, otherwise from a built-in color sequence. This series is deliberately kept separate from the generic theme slots: switching the document's theme should not also change category colors. Its defaults are all eight colors of ColorBrewer "Dark2"; anyone wanting different category colors simply declares `semcat-categories` again after loading `semcat`, via `\DeclareOsgColorSeries`. The `slot` key picks a slot of the active theme palette explicitly – but only while `osgstyler` is loaded; without it, the built-in rotation applies instead. `index=` selects a particular one-based entry from `semcat-categories`; without `osgstyler`, the same index is read from the built-in color sequence.

A new qualitative color series can explicitly be generated from the active `osgstyler` theme palette with `\GenerateSemCatColorSeries`. `count` determines its length in the range 2–32. The command is a convenience wrapper around `\GenerateOsgColorSeries` from `osgstyler` and uses its active primary, accent, and background colors. The call replaces `semcat-categories` and recolors already defined categories using its automatic rotation or `index=`. Explicit `color=` and `slot=` assignments remain unchanged.

```
\GenerateSemCatColorSeries[count=8]
```

In addition to color/slot, a category can specify *how* its block-level marking (see below) renders: as a box (the default) or as a continuous margin bar. While `osgstyler` is loaded, a decoration declared there can be referenced instead – its margin component then supplies the bar, its enclosure component the box, each with its own colors/measurements:

```
\SemCatDefine{G}[title={Aside}, style=bar]
\DeclareOsgDecoration{marked-caution}{
  enclosure = { border-color=accent, fill-color=secondary }
}
\SemCatDefine{H}[title={Caution}, decoration=marked-caution]
```

A `variants` key overrides style or decoration depending on the active `osglecture` mode – on top of the unqualified default, which serves as the fallback when no active mode (or no `osglecture-modes` at all) has an entry:

```
\SemCatDefine{I}[
  title={Aside}, style=box,
  variants={ slides = {style=bar} }
]
```

So far the concept is strongly color-oriented – but printing a textbook often leaves few or no colors available. For that case, a symbol can take the category color's place, at the

start of the margin bar or in the box header: the symbol key names a registered symbol, and `\SemCatDeclareSymbol` registers further ones (the built-in ones come from `pifont`, which – unlike Unicode symbols – is available in virtually every installation). When a resolved `osgstyler` decoration has an affix component, its content/symbol key takes precedence over the category’s plain symbol key; the affix component’s position, font and gap are not evaluated yet.

```
\SemCatDeclareSymbol{pin}{\ding{43}}
\SemCatDefine{J}[title={Caution}, symbol=warning]
```

2.2 Marking

`\SemCat` marks text inline with the category color – independent of the category’s chosen style, since inline text cannot sensibly appear as a box or a bar:

Ordinary text with a `\SemCat{B}{marked phrase}` in the middle.

For whole passages – even across multiple paragraphs and page breaks – the `SemCatMark` environment marks the content according to the style resolved for the category (and the active mode). A margin bar is not drawn via the output routine the way the former, `changebar`-based predecessor did, but line by line via a `LuaLATEX` callback – so it survives page breaks without special handling:

```
\begin{SemCatMark}[category=B]
  Supplementary material, with a footnote\footnote{like this one},
  spanning as many paragraphs as needed.
\end{SemCatMark}
```

2.3 Explanation box

Unlike `\SemCat` and `SemCatMark`, which always stay *within* the reading flow, `SemCatExplain` is a genuine, numbered and captioned floating environment for content *outside* the reading flow: skippable supplementary explanations. It is structurally always a box, regardless of the style otherwise chosen for the category, and uses the category color only for visual identity:

```
\begin{SemCatExplain}[category=B]{Why this matters}
  Supplementary background that a reader could skip.
\end{SemCatExplain}
```

The environment is built on `float` and registers the `semcatexplain` float type (default placement `tbp`, list extension `loe`); `\listof{semcatexplain}{...}` produces a list of all explanation boxes.

2.4 QR-code annotations

`\SemCatQR` places a QR code with a caption, by default as a margin note, inline with the star form:

```
\SemCatQR[B]{https://example.org}{Further reading}
\SemCatQR*[B]{https://example.org}{Further reading}
```

`\SemCatQRRedirect` instead builds the target URL from a base URL set via `\SemCatSetRedirectBase` and a document/entry id; under Lua^AT_EX both parts are robustly URL-encoded:

```
\SemCatSetRedirectBase{https://example.org/redirect.php}
\SemCatQRRedirect[B]{42}{Further reading}
```

2.5 Mode and tagging integration

Every marking command accepts a leading angle-bracket specification that – while `osglecture-modes` is loaded – is forwarded to `\IfLectureModeTF`; without `osglecture-modes` it is ignored and the markup always appears:

```
\SemCat<article>[B]{printed-only remark}
```

`SemCatMark` and `SemCatExplain` instead take the mode via the mode key, since as `environ` environments they cannot accept angle-bracket syntax; without a mode key, the markup always appears, regardless of active modes:

```
\begin{SemCatMark}[mode=slides,category=B]
  Slides-only supplementary passage.
\end{SemCatMark}
```

When a document is tagged via `\DocumentMetadata{tagging=on}`, every category registers its own structure names (`semcat/<key>` as `Span`, `semcat/<key>/note` as `Note`), and inline marking and `SemCatMark` are tagged accordingly – both stay *within* the reading flow. `SemCatExplain` instead uses the single, category-independent structure `semcat/explain` with role `Aside`, since it deliberately sits *outside* the reading flow. QR codes themselves stay untagged, since `qrcode` draws them via its own `tikz` graphic, which does not nest cleanly inside a manually opened tag structure; the caption next to it is tagged normally as ordinary flow text.

3 Implementation

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

Index

D		osglecture (bundle)	1
<code>\DeclareOsgColorSeries</code>	2	osglecture-modes (package)	1 f., 4
<code>\DocumentMetadata</code>	1, 4	osgstyler (package)	2
F		P	
float (package)	3	pifont (package)	3
G		Q	
<code>\GenerateOsgColorSeries</code>	2	qrcode (package)	4
<code>\GenerateSemCatColorSeries</code>	2	S	
I		<code>\SemCat</code>	3
<code>\IfLectureModeTF</code>	4	<code>\SemCatDeclareSymbol</code>	3
L		<code>\SemCatDefine</code>	1
<code>\listof</code>	3	<code>\SemCatQR</code>	4
O		<code>\SemCatQRRedirect</code>	4
		<code>\SemCatSetRedirectBase</code>	4