

# Guide: Transition Engine

## 1 Introduction

The `render-transition` function is designed to automatically generate “roadmap” or “summary” slides when the document structure changes (e.g., entering a new section). It is typically invoked within a `show` heading rule.

## 2 Function documentation

`render-transition(h, transitions: ({}), mapping: ({}), ...)`

### 2.1 Parameters Reference

Option	Type	Description
<code>h</code>	heading	<b>Mandatory.</b> The heading object intercepted by the <code>show</code> rule.
<code>slide-func</code>	function	<b>Mandatory.</b> A callback <code>(fill: color, body: content) =&gt; content</code> used to create the slide. It should wrap the presentation engine's slide function (e.g., <code>polylux.slide</code> ).
<code>transitions</code>	dict	Detailed configuration for the transition engine (see below).
<code>mapping</code>	dict	Maps heading levels to roles. Example: <code>(section: 1, subsection: 2)</code> .
<code>theme-colors</code>	dict	Dictionary containing primary and accent colors used for the transition slide style.
<code>show-heading-numbering</code>	bool	Whether to display heading numbers in the roadmap. Default: <code>true</code> .
<code>numbering-format</code>	string   auto	Typst numbering format string. If <code>auto</code> , uses the document's global heading numbering. Default: <code>auto</code> .
<code>base-text-size</code>	length   auto	Base font size for the roadmap text. Default: <code>auto</code> .
<code>base-text-font</code>	string   auto	Font family for the roadmap text. Default: <code>auto</code> .

### 2.2 The transitions dictionary

This parameter allows fine-tuning the behavior and appearance of transition slides.

Key	Type	Description
<code>enabled</code>	bool	Global switch for transitions. Default: <code>true</code> .
<code>max-level</code>	int	The maximum heading level that triggers a transition. Default: 3.
<code>background</code>	color   string	Background type: "theme" (uses primary color), "none", or an explicit color.
<code>filter</code>	function	A callback <code>(heading) =&gt; bool</code> to programmatically enable/disable transitions for specific headings.
<code>style</code>	dict	Controls typography: <code>inactive-opacity</code> (default 0.3), <code>completed-opacity</code> (default 0.6), <code>active-weight</code> (default "bold").

Key	Type	Description
sections	dict	Override for section-level transitions. Contains enabled, visibility (dict), and background.
subsections	dict	Override for subsection-level transitions. Contains enabled, visibility (dict), and background.

### 2.2.1 Visibility Logic

For each transition role (parts, sections, subsections), you can define which hierarchy levels are visible using the `visibility` key:

- `"all"`: Show all headings at this level.
- `"current"`: Only show the active heading at this level.
- `"current-parent"`: Show siblings of the active heading.
- `"none"`: Hide this level.

## 3 Basic Usage

In a typical presentation, you would set up the transition logic like this:

```
#import "@preview/navigator:0.1.0": render-transition

// 1. Define a wrapper for your slide engine
#let my-slide-func(fill: white, body) = {
  set page(fill: fill)
  polylux-slide(body)
}

// 2. Apply the transition engine via a show rule
#show heading: h => {
  if h.level > 2 { return h } // Only levels 1 and 2

  render-transition(
    h,
    mapping: (section: 1, subsection: 2),
    theme-colors: (primary: blue, accent: orange),
    slide-func: my-slide-func,
  )
}
```

## 4 Advanced Customization

### 4.1 Filtering transitions

You can prevent specific headings from triggering a transition slide using the `filter` parameter.

```
#show heading: h => render-transition(
  h,
  ...,
  transitions: (
    // Only headings containing "Part" will trigger a transition
    filter: h => h.body.text.contains("Part")
  )
)
```

### 4.2 Selective visibility

You can configure different roadmap layouts for sections and subsections.

```
#show heading: h => render-transition(
  h,
  ...,
  transitions: (
    // Sections show the full plan
    sections: (visibility: (section: "all", subsection: "none")),
    // Subsections show only the current section and its children
  )
)
```

```
    subsections: (visibility: (section: "current", subsection: "current-parent")),  
  )  
)
```