

- dash-period** (number)  
The length of one dash together with whitespace. If negative, no line is drawn at all.
- style** (symbol)  
This setting determines in what style a grob is typeset. Valid choices depend on the `stencil` callback reading this property.
- thickness** (number)  
For grobs made up of lines, this is the thickness of the line. For slurs and ties, this is the distance between the two arcs of the curve's outline at its thickest point, not counting the diameter of the virtual "pen" that draws the arcs. This property is expressed as a multiple of the current staff-line thickness (i.e. the visual output is influenced by changes to `Staff.StaffSymbol.thickness`).
- zigzag-length** (dimension, in staff space)  
The length of the lines of a zigzag, relative to `zigzag-width`. A value of 1 gives 60-degree zigzags.
- zigzag-width** (dimension, in staff space)  
The width of one zigzag squiggle. This number is adjusted slightly so that the glissando line can be constructed from a whole number of squiggles.

This grob interface is used in the following graphical object(s): Section 3.1.40 [DynamicTextSpanner], page 424, Section 3.1.41 [Episema], page 426, Section 3.1.48 [Glissando], page 434, Section 3.1.52 [Hairpin], page 437, Section 3.1.53 [HorizontalBracket], page 438, Section 3.1.64 [LigatureBracket], page 453, Section 3.1.84 [OttavaBracket], page 474, Section 3.1.90 [PianoPedalBracket], page 482, Section 3.1.124 [TextSpanner], page 517, Section 3.1.131 [TrillSpanner], page 526, Section 3.1.132 [TupletBracket], page 528, Section 3.1.139 [VoiceFollower], page 536, and Section 3.1.140 [VoltaBracket], page 537.

### 3.2.62 line-spanner-interface

Generic line drawn between two objects, e.g., for use with glissandi.

`bound-details` is a nested alist. It's possible to specify settings for the sub-properties: `left`, `left-broken`, `right` and `right-broken`.

Values for the following keys may be set:

- Y** Sets the Y-coordinate of the end point, in staff-spaces offset from the staff center line. By default, it is the center of the bound object, so a glissando points to the vertical center of the note head. For horizontal spanners, such as text spanners and trill spanners, it is hardcoded to 0.

**attach-dir**

Determines where the line starts and ends in the X-direction, relative to the bound object. So, a value of -1 (or LEFT) makes the line start/end at the left side of the note head it is attached to.

- X** This is the absolute X-coordinate of the end point. Usually computed on the fly.

**stencil** Line spanners may have symbols at the beginning or end, which is contained in this sub-property. For internal use.

**text** This is a markup that is evaluated to yield the stencil.

`stencil-align-dir-y`

`stencil-offset`

Without setting one of these, the stencil is simply put at the end-point, centered on the line, as defined by the X and Y sub-properties. Setting either `stencil-align-dir-y` or `stencil-offset` will move the symbol at the edge vertically relative to the end point of the line

`arrow` Produces an arrowhead at the end-points of the line.

`padding` Controls the space between the specified end point of the line and the actual end. Without padding, a glissando would start and end in the center of each note head.

### User settable properties:

`bound-details` (list)

An alist of properties for determining attachments of spanners to edges.

`extra-dy` (number)

Slope glissandi this much extra.

`gap` (dimension, in staff space)

Size of a gap in a variable symbol.

`left-bound-info` (list)

An alist of properties for determining attachments of spanners to edges.

`right-bound-info` (list)

An alist of properties for determining attachments of spanners to edges.

`simple-Y` (boolean)

Should the Y placement of a spanner disregard changes in system heights?

`thickness` (number)

For grobs made up of lines, this is the thickness of the line. For slurs and ties, this is the distance between the two arcs of the curve's outline at its thickest point, not counting the diameter of the virtual "pen" that draws the arcs. This property is expressed as a multiple of the current staff-line thickness (i.e. the visual output is influenced by changes to `Staff.StaffSymbol.thickness`).

`to-barline` (boolean)

If true, the spanner will stop at the bar line just before it would otherwise stop.

### Internal properties:

`note-columns` (array of grobs)

An array of `NoteColumn` grobs.

This grob interface is used in the following graphical object(s): Section 3.1.40 [DynamicTextSpanner], page 424, Section 3.1.41 [Episema], page 426, Section 3.1.48 [Glissando], page 434, Section 3.1.124 [TextSpanner], page 517, Section 3.1.131 [TrillSpanner], page 526, and Section 3.1.139 [VoiceFollower], page 536.

#### 3.2.63 lyric-extender-interface

The extender is a simple line at the baseline of the lyric that helps show the length of a melisma (a tied or slurred note).