RUBY/TK - SCALE WIDGET

Description:

A **Scale** is a widget that displays a rectangular trough and a small slider. The trough corresponds to a range of real values *determinedbythefrom*, *to*, *andresolutionoptions*, and the position of the slider selects a particular real value.

Three annotations may be displayed in a scale widget:

- A label appearing at the top right of the widget *topleftforhorizontalscales*.
- A number displayed just to the left of the slider *justabovethesliderforhorizontalscales*.
- A collection of numerical tick marks just to the left of the current value *justbelowthetroughforhorizontalscales*.

Each of these three annotations may be enabled or disabled using the configuration options.

Syntax:

Here is a simple syntax to create this widget:

```
TkScale.new{
  .....Standard Options....
  .....Widget-specific Options....
}
```

Standard Options:

- activebackground
- background
- borderwidth
- cursor
- font
- foreground
- highlightbackground
- highlightcolor
- highlightthickness
- orient
- relief
- repeatdelay
- repeatinterval
- takefocus
- troughcolor

These options have been described in previous chapter.

Widget-specific Options:

SN	Options with Description
1	**bigincrement** =>Integer Some interactions with the scale cause its value to change by *large* increments; this option specifies the size of the large increments. If specified as 0, the large increments default to 1/10 the range of the scale.
2	**command** =>String Specifies the prefix of a Ruby/Tk callback to invoke whenever the scale's value is changed via a method.
3	**digits** =>Integer An integer specifying how many significant digits should be retained when converting the value of the scale to a string. If the number is less than or equal to zero, then the scale picks the smallest value that guarantees that every possible slider position prints as a different string.
4	**from** =>Integer A real value corresponding to the left or top end of the scale.
5	**label** =>String A string to display as a label for the scale. For vertical scales the label is displayed just to the right of the top end of the scale. For horizontal scales the label is displayed just above the left end of the scale.
6	**length** =>Integer Specifies the desired long dimension of the scale in screen units
7	**resolution** =>Integer A real value specifying the resolution for the scale. If this value is greater than zero then the scale's value will always be rounded to an even multiple of this value, as will tick marks and the endpoints of the scale. If the value is less than zero then no rounding occurs. Defaults to 1
8	**showvalue** =>Boolean Specifies a boolean value indicating whether or not the current value of the scale is to be displayed.
9	**sliderlength** =>Integer Specfies the size of the slider, measured in screen units along the slider's long dimension.
10	**sliderrelief** =>String Specifies the relief to use when drawing the slider, such as **raised** or **sunken**.
11	**state** =>String Specifies one of three states for the scale: **normal**, **active**, or **disabled**.

12 **tickinterval** =>Integer

Must be a real value. Determines the spacing between numerical tick marks displayed below or to the left of the slider. If 0, no tick marks will be displayed.

13 **to** =>Integer

Specifies a real value corresponding to the right or bottom end of the scale. This value may be either less than or greater than the **from** option.

14 **variable** =>Variable

Specifies the name of a global variable to link to the scale. Whenever the value of the variable changes, the scale will update to reflect this value. Whenever the scale is manipulated interactively, the variable will be modified to reflect the scale's new value.

15 **width** =>Integer

Specifies the desired narrow dimension of the trough in screen units

Manipulating Scales:

The following methods are available for scale widgets:

- **coords***?value?* Returns a list whose elements are the x and y coordinates of the point along the centerline of the trough that corresponds to value. If value is omitted then the scale's current value is used.
- **get***?x, y?* If x and y are omitted, returns the current value of the scale. If x and y are specified, they give pixel coordinates within the widget; the command returns the scale value corresponding to the given pixel.
- **identify***x, y* Returns a string indicating what part of the scale lies under the coordinates given by *x* and *y*. A return value of **slider** means that the point is over the slider; **trough1** means that the point is over the portion of the slider above or to the left of the slider; and **trough2** means that the point is over the portion of the slider below or to the right of the slider.
- **set***value* This command is invoked to change the current value of the scale, and hence the position at which the slider is displayed. Value gives the new value for the scale. The command has no effect if the scale is disabled.

Event Bindings:

Ruby/Tk automatically creates class bindings for scales that give them the following default behavior. Where the behavior is different for vertical and horizontal scales, the horizontal behavior is described in parentheses.

- If button 1 is pressed in the trough, the scale's value will be incremented or decremented by the value of the resolution option so that the slider moves in the direction of the cursor. If the button is held down, the action auto-repeats.
- If button 1 is pressed over the slider, the slider can be dragged with the mouse.
- If button 1 is pressed in the trough with the Control key down, the slider moves all the way to the end of its range, in the direction towards the mouse cursor.
- If button 2 is pressed, the scale's value is set to the mouse position. If the mouse is dragged with button 2 down, the scale's value changes with the drag.
- The Up and Left keys move the slider up *left* by the value of the resolution option.
- The Down and Right keys move the slider down *right* by the value of the resolution option.

- Control-Up and Control-Left move the slider up *left* by the value of the bigIncrement option.
- Control-Down and Control-Right move the slider down *right* by the value of the bigIncrement option.
- Home moves the slider to the top *left* end of its range.
- End moves the slider to the bottom *right* end of its range.

If the scale is disabled using the state option, then none of the above bindings have any effect.

Examples:

```
require "tk"

$scale = TkScale.new {
  orient 'horizontal'
  length 280
  from 0
  to 250
  command (proc {printheight})
  tickinterval 50
  pack
}

def printheight
  height = $scale.get()
  print height, "\n"
end

Tk.mainloop
```

This will produce the following result: