NAME

options − Standard options supported by widgets

DESCRIPTION

This manual entry describes the common configuration options supported by widgets in the Tk toolkit. Every widget does not necessarily support every option (see the manual entries for individual widgets for a list of the standard options supported by that widget), but if a widget does support an option with one of the names listed below, then the option has exactly the effect described below.

In the descriptions below, “Command-Line Name” refers to the switch used in class commands and **configure** widget commands to set this value. For example, if an option’s command-line switch is **−foreground** and there exists a widget **.a.b.c**, then the command

```
.a.b.c   configure   −foreground black
```

may be used to specify the value **black** for the option in the widget **.a.b.c**. Command-line switches may be abbreviated, as long as the abbreviation is unambiguous. “Database Name” refers to the option’s name in the option database (e.g. in .Xdefaults files). “Database Class” refers to the option’s class value in the option database.

Command-Line Name:	**−activebackground**
Database Name:	**activeBackground**
Database Class:	**Foreground**

Specifies background color to use when drawing active elements. An element (a widget or portion of a widget) is active if the mouse cursor is positioned over the element and pressing a mouse button will cause some action to occur. If strict Motif compliance has been requested by setting the **tk_strictMotif** variable, this option will normally be ignored; the normal background color will be used instead. For some elements on Windows and Macintosh systems, the active color will only be used while mouse button 1 is pressed over the element.

Command-Line Name:	**−activeborderwidth**
Database Name:	**activeBorderWidth**
Database Class:	**BorderWidth**

Specifies a non-negative value indicating the width of the 3-D border drawn around active elements. See above for definition of active elements. The value may have any of the forms acceptable to **Tk_GetPixels**. This option is typically only available in widgets displaying more than one element at a time (e.g. menus but not buttons).

Command-Line Name:	**−activeforeground**
Database Name:	**activeForeground**
Database Class:	**Background**

Specifies foreground color to use when drawing active elements. See above for definition of active elements.

Command-Line Name:	**−anchor**
Database Name:	**anchor**
Database Class:	**Anchor**

Specifies how the information in a widget (e.g. text or a bitmap) is to be displayed in the widget. Must be one of the values **n**, **ne**, **e**, **se**, **s**, **sw**, **w**, **nw**, or **center**. For example, **nw** means display the information such that its top-left corner is at the top-left corner of the widget.

Command-Line Name:	**−background or −bg**
Database Name:	**background**
Database Class:	**Background**

Specifies the normal background color to use when displaying the widget.

Command-Line Name:	**−bitmap**
Database Name:	**bitmap**
Database Class:	**Bitmap**

Specifies a bitmap to display in the widget, in any of the forms acceptable to **Tk_GetBitmap**. The exact way in which the bitmap is displayed may be affected by other options such as **anchor** or **justify**. Typically, if this option is specified then it overrides other options that specify a textual value to display in the widget but this is controlled by the **compound** option; the **bitmap** option may be reset to an empty string to re-enable a text display. In widgets that support both **bitmap** and **image** options, **image** will usually override **bitmap**.

Command-Line Name:	**−borderwidth or −bd**
Database Name:	**borderWidth**
Database Class:	**BorderWidth**

Specifies a non-negative value indicating the width of the 3-D border to draw around the outside of the widget (if such a border is being drawn; the **relief** option typically determines this). The value may also be used when drawing 3-D effects in the interior of the widget. The value may have any of the forms acceptable to **Tk_GetPixels**.

Command-Line Name:	**−cursor**
Database Name:	**cursor**
Database Class:	**Cursor**

Specifies the mouse cursor to be used for the widget. The value may have any of the forms acceptable to **Tk_GetCursor**. In addition, if an empty string is specified, it indicates that the widget should defer to its parent for cursor specification.

Command-Line Name:	**−compound**
Database Name:	**compound**
Database Class:	**Compound**

Specifies if the widget should display text and bitmaps/images at the same time, and if so, where the bitmap/image should be placed relative to the text. Must be one of the values **none**, **bottom**, **top**, **left**, **right**, or **center**. For example, the (default) value **none** specifies that the bitmap or image should (if defined) be displayed instead of the text, the value **left** specifies that the bitmap or image should be displayed to the left of the text, and the value **center** specifies that the bitmap or image should be displayed on top of the text.

Command-Line Name:	**−disabledforeground**
Database Name:	**disabledForeground**
Database Class:	**DisabledForeground**

Specifies foreground color to use when drawing a disabled element. If the option is specified as an empty string (which is typically the case on monochrome displays), disabled elements are drawn with the normal foreground color but they are dimmed by drawing them with a stippled fill pattern.

Command-Line Name:	**−exportselection**
Database Name:	**exportSelection**
Database Class:	**ExportSelection**

Specifies whether or not a selection in the widget should also be the X selection. The value may have any of the forms accepted by **Tcl_GetBoolean**, such as **true**, **false**, **0**, **1**, **yes**, or **no**. If the selection is exported, then selecting in the widget deselects the current X selection, selecting outside the widget deselects any widget selection, and the widget will respond to selection retrieval requests when it has a selection. The default is usually for widgets to export selections.

Command-Line Name:	**−font**
Database Name:	**font**
Database Class:	**Font**

Specifies the font to use when drawing text inside the widget. The value may have any of the forms described in the **font** manual page under **FONT DESCRIPTION**.

Command-Line Name:	**−foreground or −fg**
Database Name:	**foreground**
Database Class:	**Foreground**

Specifies the normal foreground color to use when displaying the widget.

Command-Line Name:	**−highlightbackground**
Database Name:	**highlightBackground**
Database Class:	**HighlightBackground**

Specifies the color to display in the traversal highlight region when the widget does not have the input focus.

Command-Line Name:	**−highlightcolor**
Database Name:	**highlightColor**
Database Class:	**HighlightColor**

Specifies the color to use for the traversal highlight rectangle that is drawn around the widget when it has the input focus.

Command-Line Name:	**−highlightthickness**
Database Name:	**highlightThickness**
Database Class:	**HighlightThickness**

Specifies a non-negative value indicating the width of the highlight rectangle to draw around the outside of the widget when it has the input focus. The value may have any of the forms acceptable to **Tk_GetPixels**. If the value is zero, no focus highlight is drawn around the widget.

Command-Line Name:	**−image**
Database Name:	**image**
Database Class:	**Image**

Specifies an image to display in the widget, which must have been created with the **image create** command. Typically, if the **image** option is specified then it overrides other options that specify a bitmap or textual value to display in the widget, though this is controlled by the **compound** option; the **image** option may be reset to an empty string to re-enable a bitmap or text display.

Command-Line Name:	**−insertbackground**
Database Name:	**insertBackground**
Database Class:	**Foreground**

Specifies the color to use as background in the area covered by the insertion cursor. This color will normally override either the normal background for the widget (or the selection background if the insertion cursor happens to fall in the selection).

Command-Line Name:	**−insertborderwidth**
Database Name:	**insertBorderWidth**
Database Class:	**BorderWidth**

Specifies a non-negative value indicating the width of the 3-D border to draw around the insertion cursor. The value may have any of the forms acceptable to **Tk_GetPixels**.

Command-Line Name:	**−insertofftime**
Database Name:	**insertOffTime**
Database Class:	**OffTime**

Specifies a non-negative integer value indicating the number of milliseconds the insertion cursor should remain “off” in each blink cycle. If this option is zero then the cursor does not blink: it is on all the time.

Command-Line Name:	**−insertontime**
Database Name:	**insertOnTime**
Database Class:	**OnTime**

Specifies a non-negative integer value indicating the number of milliseconds the insertion cursor should remain “on” in each blink cycle.

Command-Line Name:	**−insertwidth**
Database Name:	**insertWidth**

Database Class:	**InsertWidth**

Specifies a value indicating the total width of the insertion cursor. The value may have any of the forms acceptable to **Tk_GetPixels**. If a border has been specified for the insertion cursor (using the **insertBorderWidth** option), the border will be drawn inside the width specified by the **insertWidth** option.

Command-Line Name:	**−jump**
Database Name:	**jump**
Database Class:	**Jump**

For widgets with a slider that can be dragged to adjust a value, such as scrollbars, this option determines when notifications are made about changes in the value. The option’s value must be a boolean of the form accepted by **Tcl_GetBoolean**. If the value is false, updates are made continuously as the slider is dragged. If the value is true, updates are delayed until the mouse button is released to end the drag; at that point a single notification is made (the value “jumps” rather than changing smoothly).

Command-Line Name:	**−justify**
Database Name:	**justify**
Database Class:	**Justify**

When there are multiple lines of text displayed in a widget, this option determines how the lines line up with each other. Must be one of **left**, **center**, or **right**. **Left** means that the lines’ left edges all line up, **center** means that the lines’ centers are aligned, and **right** means that the lines’ right edges line up.

Command-Line Name:	**−orient**
Database Name:	**orient**
Database Class:	**Orient**

For widgets that can lay themselves out with either a horizontal or vertical orientation, such as scrollbars, this option specifies which orientation should be used. Must be either **horizontal** or **vertical** or an abbreviation of one of these.

Command-Line Name:	**−padx**
Database Name:	**padX**
Database Class:	**Pad**

Specifies a non-negative value indicating how much extra space to request for the widget in the X-direction. The value may have any of the forms acceptable to **Tk_GetPixels**. When computing how large a window it needs, the widget will add this amount to the width it would normally need (as determined by the width of the things displayed in the widget); if the geometry manager can satisfy this request, the widget will end up with extra internal space to the left and/or right of what it displays inside. Most widgets only use this option for padding text: if they are displaying a bitmap or image, then they usually ignore padding options.

Command-Line Name:	**−pady**
Database Name:	**padY**
Database Class:	**Pad**

Specifies a non-negative value indicating how much extra space to request for the widget in the Y-direction. The value may have any of the forms acceptable to **Tk_GetPixels**. When computing how large a window it needs, the widget will add this amount to the height it would normally need (as determined by the height of the things displayed in the widget); if the geometry manager can satisfy this request, the widget will end up with extra internal space above and/or below what it displays inside. Most widgets only use this option for padding text: if they are displaying a bitmap or image, then they usually ignore padding options.

Command-Line Name:	**−relief**
Database Name:	**relief**
Database Class:	**Relief**

Specifies the 3-D effect desired for the widget. Acceptable values are **raised**, **sunken**, **flat**, **ridge**, **solid**, and **groove**. The value indicates how the interior of the widget should appear relative to its exterior; for example, **raised** means the interior of the widget should appear to protrude from the screen, relative to the exterior of the widget.

Command-Line Name: **−repeatdelay**
Database Name: **repeatDelay**
Database Class: **RepeatDelay**

Specifies the number of milliseconds a button or key must be held down before it begins to auto-repeat. Used, for example, on the up- and down-arrows in scrollbars.

Command-Line Name: **−repeatinterval**
Database Name: **repeatInterval**
Database Class: **RepeatInterval**

Used in conjunction with **repeatDelay**: once auto-repeat begins, this option determines the number of milliseconds between auto-repeats.

Command-Line Name: **−selectbackground**
Database Name: **selectBackground**
Database Class: **Foreground**

Specifies the background color to use when displaying selected items.

Command-Line Name: **−selectborderwidth**
Database Name: **selectBorderWidth**
Database Class: **BorderWidth**

Specifies a non-negative value indicating the width of the 3-D border to draw around selected items. The value may have any of the forms acceptable to **Tk_GetPixels**.

Command-Line Name: **−selectforeground**
Database Name: **selectForeground**
Database Class: **Background**

Specifies the foreground color to use when displaying selected items.

Command-Line Name: **−setgrid**
Database Name: **setGrid**
Database Class: **SetGrid**

Specifies a boolean value that determines whether this widget controls the resizing grid for its top-level window. This option is typically used in text widgets, where the information in the widget has a natural size (the size of a character) and it makes sense for the window's dimensions to be integral numbers of these units. These natural window sizes form a grid. If the **setGrid** option is set to true then the widget will communicate with the window manager so that when the user interactively resizes the top-level window that contains the widget, the dimensions of the window will be displayed to the user in grid units and the window size will be constrained to integral numbers of grid units. See the section **GRIDDED GEOMETRY MANAGEMENT** in the **wm** manual entry for more details.

Command-Line Name: **−takefocus**
Database Name: **takeFocus**
Database Class: **TakeFocus**

Determines whether the window accepts the focus during keyboard traversal (e.g., Tab and Shift-Tab). Before setting the focus to a window, the traversal scripts consult the value of the **takeFocus** option. A value of **0** means that the window should be skipped entirely during keyboard traversal. **1** means that the window should receive the input focus as long as it is viewable (it and all of its ancestors are mapped). An empty value for the option means that the traversal scripts make the decision about whether or not to focus on the window: the current algorithm is to skip the window if it is disabled, if it has no key bindings, or if it is not viewable. If the value has any other form, then the traversal scripts take the value, append the name of the window to it (with a separator space), and evaluate the resulting string as a Tcl script. The script must return **0**, **1**, or an empty string: a **0** or **1** value specifies whether the window will receive the input focus, and an empty string results in the default decision described above. Note: this interpretation of the option is defined entirely by the Tcl scripts that implement traversal: the widget implementations ignore the option entirely, so you can change its meaning if you redefine the keyboard traversal scripts.

Command-Line Name:	**−text**
Database Name:	**text**
Database Class:	**Text**

Specifies a string to be displayed inside the widget. The way in which the string is displayed depends on the particular widget and may be determined by other options, such as **anchor** or **justify**.

Command-Line Name:	**−textvariable**
Database Name:	**textVariable**
Database Class:	**Variable**

Specifies the name of a variable. The value of the variable is a text string to be displayed inside the widget; if the variable value changes then the widget will automatically update itself to reflect the new value. The way in which the string is displayed in the widget depends on the particular widget and may be determined by other options, such as **anchor** or **justify**.

Command-Line Name:	**−troughcolor**
Database Name:	**troughColor**
Database Class:	**Background**

Specifies the color to use for the rectangular trough areas in widgets such as scrollbars and scales. This option is ignored for scrollbars on Windows (native widget does not recognize this option).

Command-Line Name:	**−underline**
Database Name:	**underline**
Database Class:	**Underline**

Specifies the integer index of a character to underline in the widget. This option is used by the default bindings to implement keyboard traversal for menu buttons and menu entries. 0 corresponds to the first character of the text displayed in the widget, 1 to the next character, and so on.

Command-Line Name:	**−wraplength**
Database Name:	**wrapLength**
Database Class:	**WrapLength**

For widgets that can perform word-wrapping, this option specifies the maximum line length. Lines that would exceed this length are wrapped onto the next line, so that no line is longer than the specified length. The value may be specified in any of the standard forms for screen distances. If this value is less than or equal to 0 then no wrapping is done: lines will break only at newline characters in the text.

Command-Line Name:	**−xscrollcommand**
Database Name:	**xScrollCommand**
Database Class:	**ScrollCommand**

Specifies the prefix for a command used to communicate with horizontal scrollbars. When the view in the widget's window changes (or whenever anything else occurs that could change the display in a scrollbar, such as a change in the total size of the widget's contents), the widget will generate a Tcl command by concatenating the scroll command and two numbers. Each of the numbers is a fraction between 0 and 1, which indicates a position in the document. 0 indicates the beginning of the document, 1 indicates the end, .333 indicates a position one third the way through the document, and so on. The first fraction indicates the first information in the document that is visible in the window, and the second fraction indicates the information just after the last portion that is visible. The command is then passed to the Tcl interpreter for execution. Typically the **xScrollCommand** option consists of the path name of a scrollbar widget followed by "set", e.g. ".x.scrollbar set": this will cause the scrollbar to be updated whenever the view in the window changes. If this option is not specified, then no command will be executed.

Command-Line Name:	**−yscrollcommand**
Database Name:	**yScrollCommand**
Database Class:	**ScrollCommand**

Specifies the prefix for a command used to communicate with vertical scrollbars. This option is treated in the same way as the **xScrollCommand** option, except that it is used for vertical scrollbars and is provided by widgets that support vertical scrolling. See the description of **xScrollCommand** for details on how this option is used.

SEE ALSO

colors, cursors, font

KEYWORDS

class, name, standard option, switch