Recent Improvements in Tk 9

by

Csaba Nemethi

csaba.nemethi@t-online.de

Contents

---

1. Abstract

This talk is about the progress achieved in the last few months regarding Tk 9:

- The improved look of several Ttk widgets in the built-in themes **`default`**, **`alt`**, **`clam`**, and **`classic`**;
- Focus ring support for the widgets ttk::entry, ttk::spinbox, and ttk::combobox;
- Support for non-default ttk::notebook tab positions;
- Support for the new **`<TouchpadScroll>`** event on Windows and macOS Aqua.

---

2. Improved Look of Several Ttk Widgets

Improvements related to the arrows in Ttk widgets of the themes `default`, `alt`, `clam`, and `classic`:

- **`default`** and **`alt`** themes: The arrow boxes of the ttk::combobox and ttk::spinbox widgets now have a *visible* left border; the arrow within the ttk::combobox widget is now vertically centered; the arrow boxes of the ttk::scrollbar widget are now square-shaped.
- **`clam`** theme: The arrow boxes of the ttk::spinbox widget now have optimal dimensions, due to which the widget is no longer higher than needed and its arrows are no longer tiny; the arrrows of the ttk::scrollbar widget are now horizontally centered.

- **`classic`** theme: Replaced the ugly arrow boxes of the ttk::combobox and ttk::spinbox widgets with new ones, which are imported from the **`default`** theme (thanks to **Emiliano Gavilan**):

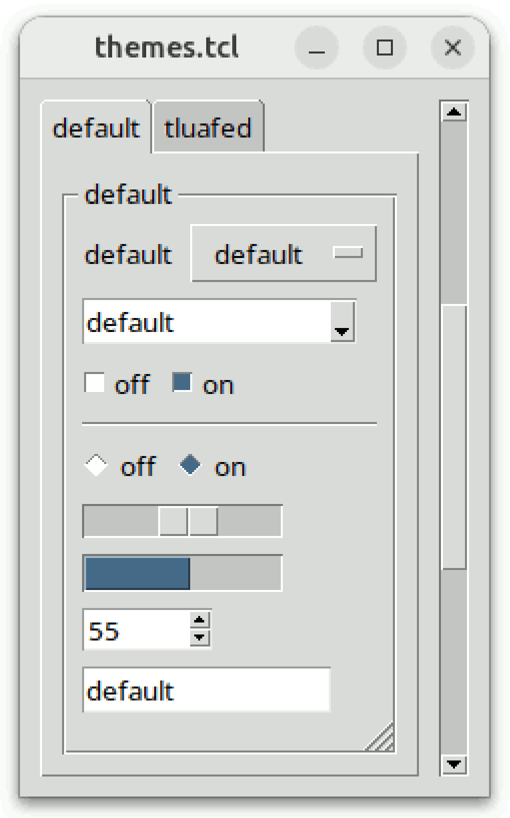
```
ttk::style element create Combobox.downarrow from default

ttk::style element create Spinbox.uparrow    from default
ttk::style element create Spinbox.downarrow  from default
```

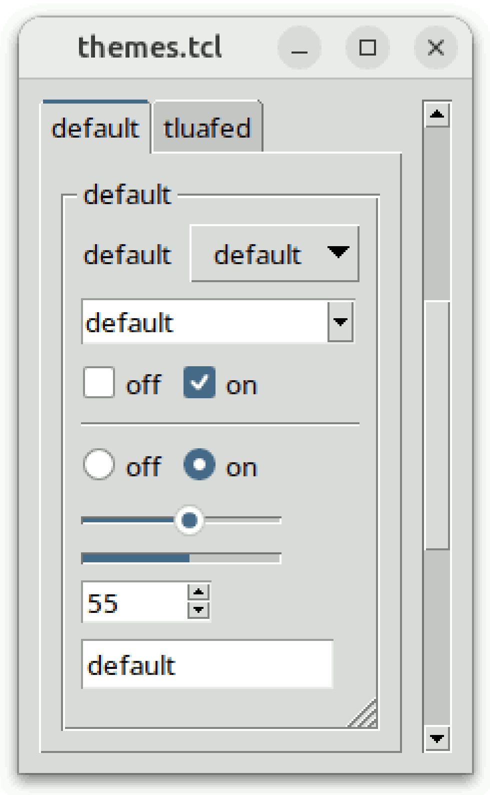
Further improvements:

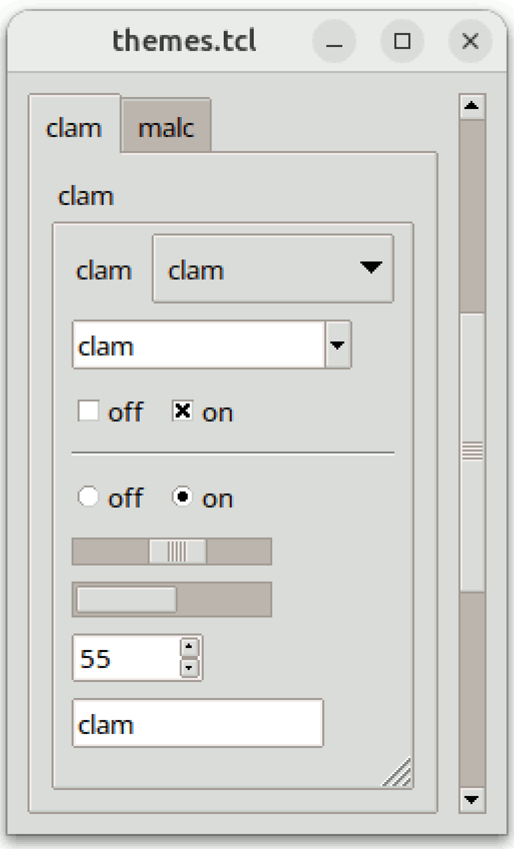
- **`default`** theme: Added a highlighting to the selected ttk::notebook tab; replaced the ttk::menubutton indicator with a more modern one, like in the **`alt`** and **`clam`** themes; the ttk::scale and ttk::progressbar widgets now have a nice modern look; changed the value of the **`-selectborderwidth`** style configuration option from `1` to `0`.

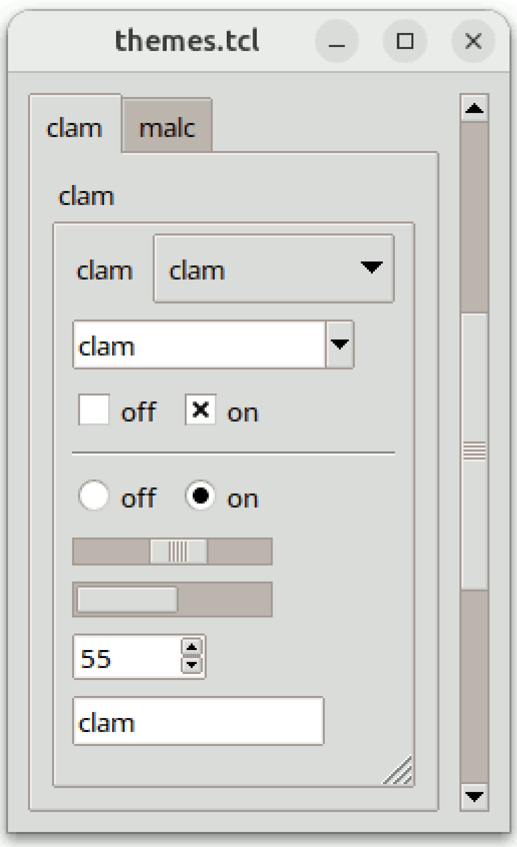
- **`classic`** theme: Changed the value of the **`-borderwidth`** and **`-troughborderwidth`** style configuration options from `2` to `1` and that of the **`-selectborderwidth`** option from `1` to `0`; changed the default width of the ttk::scrollbar widget from `11.25p` to `9p` (`15` px to `12` px on an unscaled screen); changed the value of the **`-troughcolor`** style configuration option from `"#c3c3c3"` to `"#b3b3b3"` (thanks to **Emiliano Gavilan**).

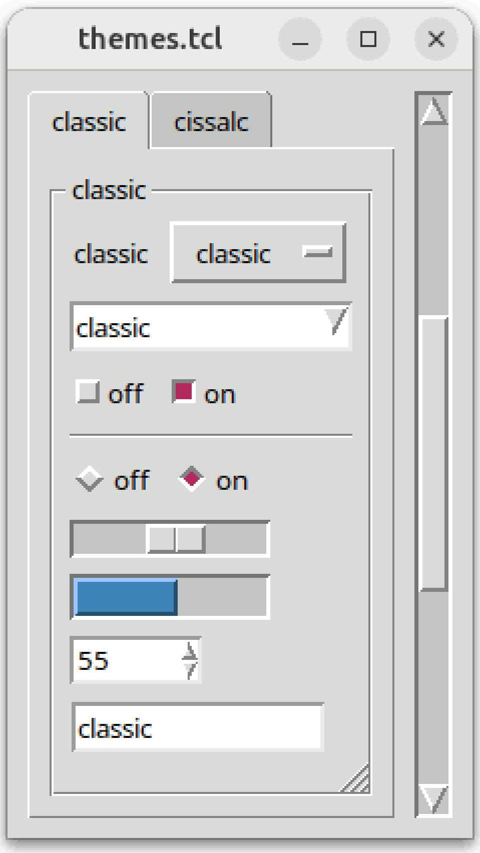
A few screenshots (Tk 8.6.14 vs Tk 9):

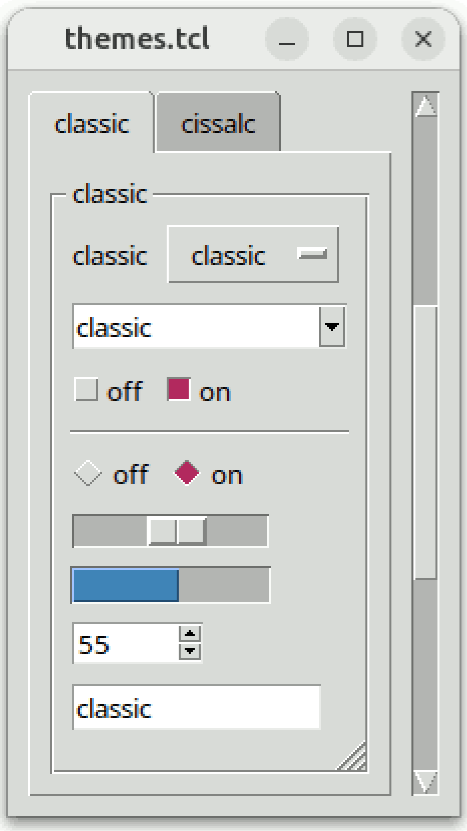





themes.tcl
clam
malc
clam
clam
clam
clam
off
on
off
on
55
clam


themes.tcl
clam
malc
clam
clam
clam
clam
off
on
off
on
55
clam


themes.tcl
classic
cissalc
classic
classic
classic
classic
off
on
off
on
55
classic


themes.tcl
classic
cissalc
classic
classic
classic
classic
off
on
off
on
55
classic

3. Focus Ring Support for Ttk Widgets

The implementation of the focus ring around some Ttk widgets is theme-specific:

- **`default`** and **`alt`** themes: Extended the implementation of the **`field`** element to support the new **`-focuswidth`** and **`-focuscolor`** options, whose values for the ttk::entry, ttk::combobox, and ttk::spinbox widgets are set at script level as follows:

```
ttk::style configure TEntry    -focuswidth 2 -focuscolor $colors(-selectbg)
ttk::style configure TCombobox -focuswidth 1 -focuscolor $colors(-selectbg)
ttk::style configure TSpinbox  -focuswidth 1 -focuscolor $colors(-selectbg)
```

- **`clam`** theme: The focus ring around the ttk::entry, ttk::combobox, and ttk::spinbox widgets is drawn by using different values for the already existing **`field`** element options **`-bordercolor`** and **`-lightcolor`**. The last two lines below were aded in Tk 9:

```
ttk::style map TEntry    -bordercolor [list focus $colors(-selectbg)] \
    -lightcolor [list focus #6f9dc6]
ttk::style map TCombobox -bordercolor [list focus $colors(-selectbg)]
ttk::style map TSpinbox  -bordercolor [list focus $colors(-selectbg)]
```

- **`classic`** theme: The focus ring is actually the **`highlight`** element, which has always been the outermost element of the themes **`TButton`**, **`TCheckbutton`**, **`TRadiobutton`**, **`TMenubutton`**, and **`TEntry`**. In Tk 9 it was added as outermost element to the following further layouts: **`TCombobox`**, **`TSpinbox`**, **`Horizontal.TScale`**, **`Vertical.TScale`**, and **`Treeview`** (thanks to **Emiliano Gavilan**).

---

4. Non-default ttk::notebook Tab Positions

Extended the C code to *properly* support not only the default value **`nw`** of the **`TNotebook`** style configuration option **`-tabposition`**, but also other values, like **`sw`**, **`wn`**, and **`en`**.

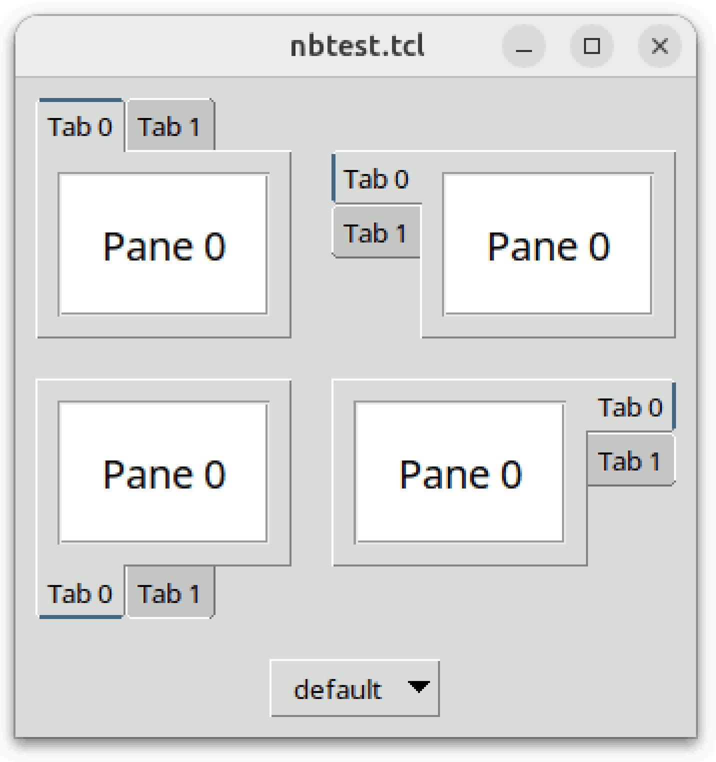
The Ttk library files `altTheme.tcl`, `clamTheme.tcl`, `vistaTheme.tcl`, `xpTheme.tcl`, and `winTheme.tcl` contain the settings for the styles **`TNotebook`** and **`TNotebook.Tab`**, corresponding to the default tab position **`nw`**:

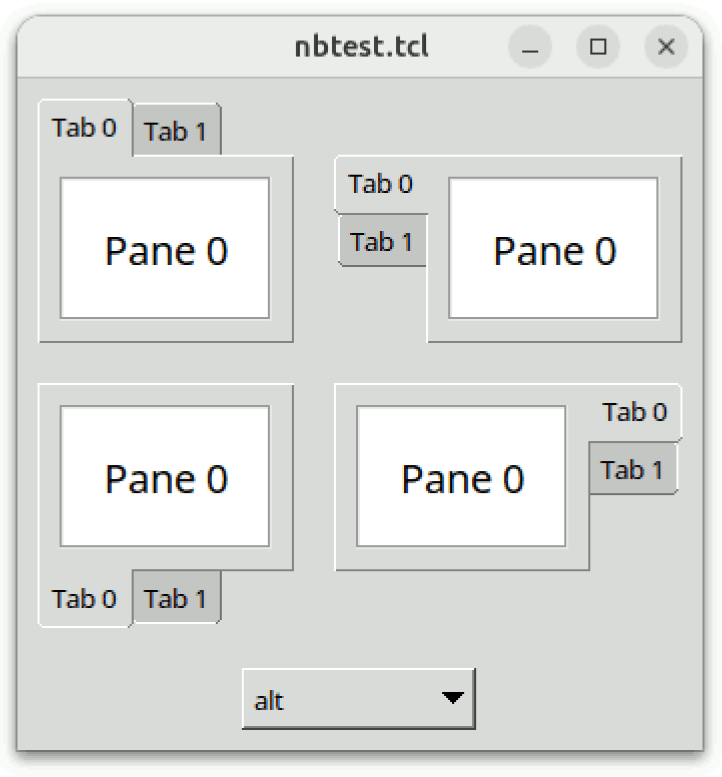
```
ttk::style theme setting alt {
    ttk::style configure TNotebook -tabmargins     {1.5p 1.5p 0.75p 0}
    ttk::style map TNotebook.Tab -expand {selected {1.5p 1.5p 0.75p 0}}
}
ttk::style theme setting clam {
    ttk::style configure TNotebook.Tab -padding     {4.5p 1.5p 4.5p 1.5p}
    ttk::style map TNotebook.Tab -padding {selected {4.5p 3p   4.5p 1.5p}}
}
ttk::style theme setting vista|xpnative {
    ttk::style configure TNotebook -tabmargins     {2 2 2 0}
    ttk::style map TNotebook.Tab -expand {selected {2 2 2 2}}
}
ttk::style theme setting winnative {
    ttk::style configure TNotebook -tabmargins     {2 2 2 0}
    ttk::style map TNotebook.Tab -expand {selected {2 2 2 0}}
}
```

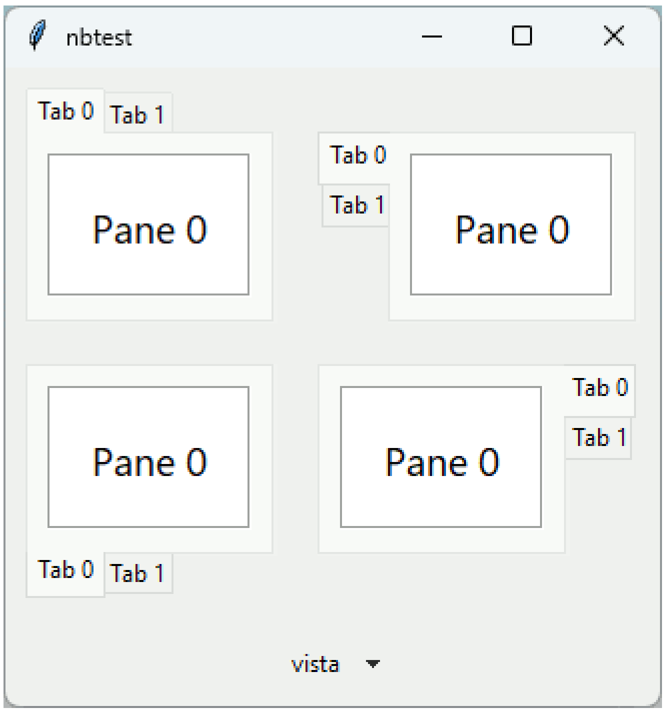
For a non-default tab position the extended C code assumes that accordingly modified settings are provided, like in the following example:

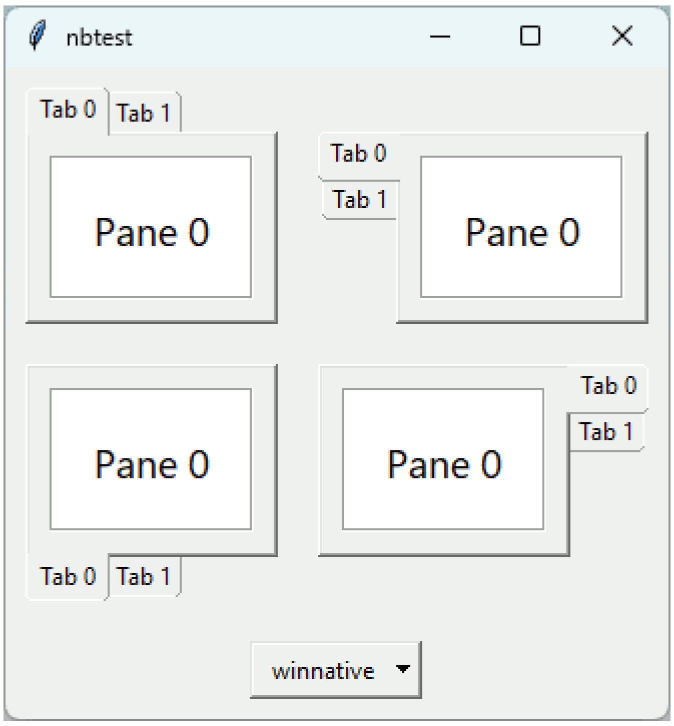
```
set themeList [ttk::style theme names]

#
# Tab position sw
#
set nbStyle SW.TNotebook
ttk::style configure $nbStyle -tabposition sw
ttk::style theme setting alt {
    ttk::style configure $nbStyle -tabmargins     {1.5p 0 0.75p 1.5p}
    ttk::style map $nbStyle.Tab -expand {selected {1.5p 0 0.75p 1.5p}}
}
ttk::style theme setting clam {
    ttk::style configure $nbStyle.Tab -padding     {4.5p 1.5p 4.5p 1.5p}
    ttk::style map $nbStyle.Tab -padding {selected {4.5p 1.5p 4.5p 3p  }}
}
foreach theme {vista xpnative} {
    if {$theme in $themeList} {
        ttk::style theme setting $theme {
            ttk::style configure $nbStyle -tabmargins     {2 0 2 2}
            ttk::style map $nbStyle.Tab -expand {selected {2 2 2 2}}
        }
    }
}
if {"winnative" in $themeList} {
    ttk::style theme setting winnative {
        ttk::style configure $nbStyle -tabmargins     {2 0 2 2}
        ttk::style map $nbStyle.Tab -expand {selected {2 0 2 2}}
    }
}
ttk::notebook .nbSW -style $nbStyle
...
```

A few screenshots:




nbtest.tcl
Tab 0
Tab 1
Pane 0
Tab 0
Tab 1
Pane 0
Pane 0
Tab 0
Tab 1
Pane 0
Tab 0
Tab 1
alt


nbtest
Tab 0
Tab 1
Pane 0
Tab 0
Tab 1
Pane 0
Pane 0
Tab 0
Tab 1
Pane 0
Tab 0
Tab 1
vista



5. Support for the New `<TouchpadScroll>` Event

Adding a new **`<TouchpadScroll>`** event to Tk was proposed and implemented in November 2023 by **Marc Culler** (see TIP 684).

Background: According to TIP 563 by **Harald Oehlmann** (back in 2020), when the mouse pointer is over a horizontal or vertical scrollbar, the mouse wheel scrolls the connected widget, regardless of whether the `Shift` key is down or not. That is, both the **`<MouseWheel>`** and **`<Shift-MouseWheel>`** events sent to a scrollbar would result in scrolling the connected widget in the direction given by the scrollbar's **`-orient`** option. Since two-finger touchpad scroll gestures almost always gave rise to *both* of these events, they resulted in annoying interferences, experienced especially during slow scrollings, as reported by **Nicolas Bats**.

To solve this problem, two-finger touchpad scroll gestures **on Windows and macOS Aqua** no longer generate **`<MouseWheel>`** and **`<Shift-MouseWheel>`** events, but they send **`<TouchpadScroll>`** events instead. These events store two 16 bit delta values in the integer provided by the **`%D`** substitution. These values can be unpacked by using the **`tk::PreciseScrollDeltas`** utility procedure, like in the following example:

```
lassign [tk::PreciseScrollDeltas %D] deltaX deltaY
```

The binding scripts for scrollbar widgets (see the Tk library file `scrlbar.tcl`) simply ignore the value of `deltaY` for a horizontal scrollbar and that of `deltaX` for a vertical one, thus eliminating any potential interferences.

The *`serial`* field of a **`<TouchpadScroll>`** event, accessible as the **`%#`** substitution, holds a counter which is incremented each time a **`<TouchpadScroll>`** event is generated (which happens about 60 times per second during a two-finger gesture). This allows a binding script to, for example, only respond to every 5th **`<TouchpadScroll>`** event by testing if the counter is divisible by 5.

On X11 there is currently no support for the **`<TouchpadScroll>`** event, two-finger touchpad scroll gestures continue to generate **`<MouseWheel>`** and **`<Shift-MouseWheel>`** events. To minimize the number of undesirable artifacts, the improved binding scripts for scrollbar widgets count both the **`<MouseWheel>`** and **`<Shift-MouseWheel>`** events and ignore the non-dominant ones.

Example of an extended event handling that supports both the mouse wheel and the two-finger gestures (see the Tk library file `listbox.tcl`):

```
bind Listbox <MouseWheel> {
    tk::MouseWheel %W y %D -40.0 units
}
bind Listbox <Option-MouseWheel> {
    tk::MouseWheel %W y %D -12.0 units
}
bind Listbox <Shift-MouseWheel> {
    tk::MouseWheel %W x %D -40.0 units
}
bind Listbox <Shift-Option-MouseWheel> {
    tk::MouseWheel %W x %D -12.0 units
}
bind Listbox <TouchpadScroll> {
    if {%# %% 5 != 0} {
        return
    }
    lassign [tk::PreciseScrollDeltas %D] deltaX deltaY
    if {$deltaX != 0} {
        %W xview scroll [expr {-$deltaX}] units
    }
    if {$deltaY != 0} {
        %W yview scroll [expr {-$deltaY}] units
    }
}
```

Libraries known to support the **`<TouchpadScroll>`** event: Mentry 4.0+, Scrollutil 2.0+, Tablelist 7.0+.