dgw::tlistbox - a tablelist based listbox with multiline support

Detlef Groth, Schwielowsee, Germany

2019-10-21

Name

dgw::tlistbox - a tablelist based listbox with multiline text support and a search entry.

Table of Contents

- Synopsis
- Description
- Command
- Widget Options
- Widget Commands
- Example
- Installation
- Demo
- Documentation
- See also
- Todo
- Authors
- Copyright
- License

Synopsis

```
package require dgw::tlistbox
namespace import ::dgw::tlistbox
tlistbox pathName options
pathName configure -searchentry boolean
pathName configure -browsecmd script
pathName itemconfigure index -option value
pathName itemcget index -option
pathName iteminsert index string
```

Description

tlistbox - a listbox widget based on the Csaba Nemethis tablelist widget with support for multiline text which can be wrapped. As **tlistbox** is based on the tablelist widget, it suppports the standard options and commands of tablelist. The **tlistbox** widget is a two column tablelist widget where the second column is hidden to allow invisible storage of data belonging to a **tlistbox** item or a cell in tablelist terms.

For convinience, and to make the widget more **listbox** like, a few options and methods were added in addition to the tablelist options. Beside of standard listbox functionality the user of this widget can display on top of the **tlistbox** a search entry widget in order to dynamically search the **tlistbox**.

Command

tlistbox *pathName ?options?*

> Creates and configures a new tlistbox widget using the Tk window id *pathName* and the given *options*.

Widget Options

The **tlistbox** widget is a two column widget based on the tablelist widget of C. Nemethi where the second column is hidden, but usable for data storage. It therefore supports the standard options and commands of tablelist, i.e. all of the options mentioned in the *tablelist* manual in the entries for *configure*, *rowconfigure* and *columnconfigure* are available. As in a **tlistbox** widget a row consist of only one visible tablelist cell we call this also in agreement with the manual pages of the standard listbox widget also an **item**.

The **tlistbox** widgets adds the following options in addition to the options available for the tablelist widget:

-browsecmd *script*

> Set a command if the user double clicks an entry in the listbox or presses the `Return` key if the widget has the focus. The widget path and the index of the item getting the event are appended to the script call as function arguments. So the implementation of the script should have two arguments in the parameter list as shown in the following example:

```
proc click {tbl idx} {
    puts [$tbl itemcget $idx -text]
}
tlistbox .tl -browsecmd click
```

-searchentry *boolean*

> Should a search entry widget displayed on top of the **tlistbox** widget. Default: false
>
> The user can enter text in the search widget for searching the **tlistbox** widget, pressing `Return` in the *entry* widget invokes the script supplied with the *-browsecmd* option.

Please note that the tablelist options **-listvariable** should be here a nested list, where each sublist should have at maximum only two values, the first value will be in the first visible cell, the second list value will be placed in the invisible cell. The latter can be retrieved via *pathName itemconfigure index -data.*

Widget Commands

Each **tlistbox** widgets supports the tablelist widget commands. Further two listbox like methods are implemented for convinience to configure individual cells, in listbox terms also called **items**. Please note, that you can also use in a similar way the *cellconfigure* and *cellcget* functions of the tablelist widget.

pathName **itemcget** *index option*

> Retrieves the given option for the item (first cell of the row). See *itemconfigure* for an explanation of the options.

pathName **itemconfigure** *index option value ?option value ...?*

Configure the **item** (first cell of the row) indicated by *index* with the given value. All options mentioned in *cellconfigure* of the tablelist manual can be used, such as *-text*, *-foreground*, *-background*. In addition to the tablelist cell options for the **tlistbox** widget an option *-data* is provided which allows storage of text data in the hidden second column of the *tlistbox* widget. Beside numerical indices as well *end* can be used as index.

pathName **iteminsert** *index string*

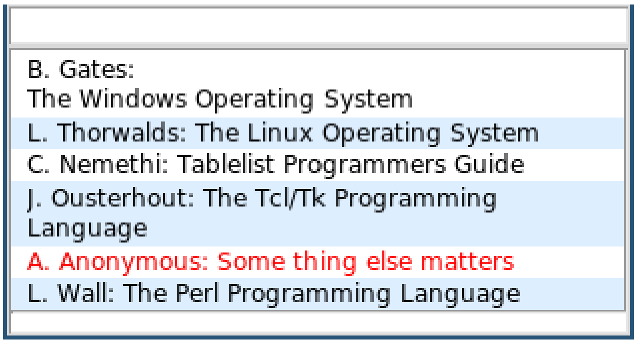
Insert the given string at position index into the **tlistbox** value. This is just a convinience function which does the same as *tablelist insert index {"string"}* But here you don't need to add add extra braces. Note, that you can not insert data text into the hidden column with this method and you can only add one element per function call. Example:

```
tlistbox .tl
.tl insert end {"Hello Text 1"}       ;# inserts all
.tl insert end "Hello Text 2"         ;# only inserts Hello
.tl iteminsert end "Hello Text 3"     ;# inserts all
```

Example

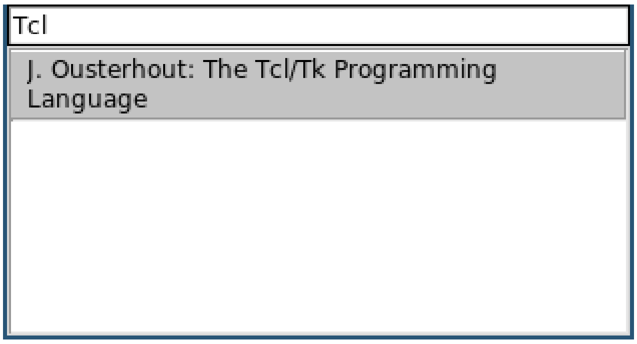
```
package require dgw::tlistbox
namespace import ::dgw::tlistbox

set data { {"B. Gates:\nThe Windows Operating System" "Hidden Data"}
      {"L. Thorwalds: The Linux Operating System"}
      {"C. Nemethi's: Tablelist Programmers Guide"}
      {"J. Ousterhout: The Tcl/Tk Programming Language"}
}
proc click {tbl idx} {
    puts [$tbl itemcget $idx -text]
}
tlistbox .tl -listvariable data -browsecmd click -searchentry true
lappend data {"A. Anonymous: Some thing else matters"}
.tl insert end {"L. Wall: The Perl Programming Language" "1987"}
pack .tl -side top -fill both -expand yes
.tl itemconfigure end -foreground red
.tl itemconfigure end -data Hello
puts "Hello? [.tl itemcget end -data] - yes!"
```



B. Gates:
The Windows Operating System
L. Thorwalds: The Linux Operating System
C. Nemethi: Tablelist Programmers Guide
J. Ousterhout: The Tcl/Tk Programming Language
A. Anonymous: Some thing else matters
L. Wall: The Perl Programming Language

Figure 1: tlistbox example



Tcl

J. Ousterhout: The Tcl/Tk Programming Language

Figure 2: tlistbox search

Installation

Installation is easy you can easily install and use this **tlistbox** package if you have a working install of:

- the snit package which can be found in tcllib
- the tablelist package which can be found on C. Nemethi's webpage

If you have those Tcl packages installed, you can either use the tlistbox package by sourcing it with: `source /path/to/tlistbox.tcl`, by copying the folder dg to a path belonging to your Tcl `$auto_path` variable or by installing it as an Tcl-module. To do this, make a copy of `tlistbox.tcl` to a file like `tlistbox-0.1.tm` and put this file into a folder named `dgw` where the parent folder belongs to your module path. You must now adapt eventually your Tcl-module path by using in your Tcl code the command: `tcl::tm::path add /parent/dir/` of the `dgw` directory. For details of the latter consult see the manual page of tcl::tm.

Alternatively there is an install option you can use as well. Try `tclsh tlistbox.tcl --install` which should perform the procedure described above automatically. This requires eventually the setting of an environment variables like if you have no write access to all your module paths. For instance on my computer I have the following entry in my `.bashrc`

```
export TCL8_6_TM_PATH=/home/groth/.local/lib/tcl8.6
```

If I execute `tclsh tlistbox.tcl --install` the file `tlistbox.tcl` will be copied to `/home/groth/.local/lib/tcl8.6/dg/tlistbox-0.1.tm` and is thereafter available for a `package require dgw::tlistbox`.

Demo

Example code for this package can be executed by running this file using the following command line:

```
$ wish tlistbox.tcl --demo
```

The example code used for this demo can be seen in the terminal by using the following command line:

```
$ wish tlistbox.tcl --code
```

Documentation

The script contains embedded the documentation in Markdown format. To extract the documentation you should use the following command line:

```
$ tclsh tlistbox.tcl --man
```

The output of this command can be used to feed a markdown document for conversion into a markdown processor for instance to convert it into an html or pdf document you could execute the following commands:

```
tclsh ../tlistbox.tcl --man > tlistbox.md
pandoc -i tlistbox.md -s -o tlistbox.html
pandoc -i tlistbox.md -s -o tlistbox.tex
pdflatex tlistbox.tex
```

See also

- tablelist man page: http://www.nemethi.de/tablelist/tablelistWidget.html

Todos

- probably as usually more documentation
- github url for putting the code

Authors

The **tlistbox** widget is based on Csaba Nemethi's great tablelist widget.

The **tlistbox** widget was written with his help by Detlef Groth, Schwielowsee, Germany.

Copyright

License

Single-column listbox widget with multiline text and search entry, version 0.1