This tool function sets the width of the selection button in character units.

1.4.36 synclistboxes - Tool

Package **display**
Module **widgets**

A collection of synchronized scrolllistboxes.
include "synclistboxes.g"

Constructors

synclistboxes	*Construct a synclistboxes*

Functions

background	*Set the background color.*
bind	*bind an event to a glish event*
borderwidth	*change the borderwidth(s)*
clear	*clear the selection at the indicated items*
delete	*delete the indicated items*
done	*Destroy ourselves*
exportselection	*export the selection to the X clipboard?*
font	*change the text font*
foreground	*Set the foreground color.*
get	*request items, i.e. map indexes to items*
height	*Set the height*
hscrollbar	*Turn on or off the horizontal scrollbars*
insert	*insert values into the listboxes*
leadbox	*which listbox is the default returned by the listbox function*
listbox	*Return an underlying listbox agent*
mode	*change listbox mode*
nearest	*return item nearest the y offset param*
relief	*change the border relief*
see	*scroll listboxes to specified index is visible*
seeoninsert	*See the end after each insert?*
select	*select the indicate items*
selection	*get indexes of the current selection*
vscrollbar	*Turn on or off the vertical scrollbar*
width	*Set the width*

Description

The synclistboxes widget is a collection of scrolllistboxes which are synchronized so that a selection in one is mirrored in all of the other listboxes and the vertical scrollbar controls all of the listboxes in the widget.
The synclistboxes widget supports all scrolllistbox widget events. However, in order to allow for fine control over the individual listboxes, many of the events have an optional additional argument which specifies the listbox which will receive that event. Each event is described below as a function due to the current limitations of the documentation system. There is only one actual function associated with this widget - listbox.
Direct access to the underlying listboxes is provided by the listbox function. This is necessary in order to support popup help on synclistboxeses. Popup help uses the tlead argument in constructing the outframe. The value of that argument needs to be one of the underlying widgets hence the need to return a reference to a true listbox. Care should be taken in using this as it allows the user to destroy the underlying listboxes and get them momentarily out of sync.

Example

```
include 'synclistboxes.g'
f := frame()
# two syncronized listboxes filled from left to right inside f
# the packing order is determined by the optional "side" argument
# to the synclistboxes function.  The listbox labels will be
# "Numbers" and "Colors".
slb := synclistboxes(f,2,"Numbers Colors");
# insert some values into the listboxes.  An equal number of values
# must be inserted into each listbox with each use of insert.
somevals := array(' ',2,9);
somevals[1,] := "one two three four five six seven eight nine";
somevals[2,] := "red orange yellow lightgrey blue brown peach mauve gold";
slb->insert(somevals);
# you decide you don't want the horizontal scrollbars
slb->hscrollbar(F)
# and you would like that listbox to only be 8 characters wide
slb->width(8, 1)
# you decide you don't want the widget to always show the end
# after each insert event
slb->seeoninsert(F)
# you want to get the current selection(s), if any
s := slb->selection();
# and you want to get all of the values at that selection
g := slb->get(s);
# g[1] will have the value from the first listbox at index s
```

```
# g[2] will have the value from the second listbox at index s
```

synclistboxes - Constructor

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Construct a synclistboxes

Synopsis
synclistboxes(parent, nboxes, labels, leadbox, side, hscrollbar, vscrollbar, seeoninsert, width, height, mode, font, relief, borderwidth, foreground, background, exportselection, fill, widgetset)

Arguments

parent	*Parent of the synclistboxes* *Allowed: Tk frame*
nboxes	*The number of listboxes this widget will contain* *Allowed: Integer* *Default: 1*
labels	*The label(s)* *Allowed: String* *Default: ''*
leadbox	*The default box returned by the listbox function* *Allowed: Integer* *Default: 1*
side	*The packing order* *Allowed: String from 'left', 'right', 'top', 'bottom'* *Default: 'left'*
hscrollbar	*Show a horizontal scrollbar in all listboxes?* *Allowed: Boolean* *Default: T*
vscrollbar	*Show a vertical scrollbar?* *Allowed: Boolean* *Default: T*
seeoninsert	*See the end after each insert?* *Allowed: Boolean* *Default: T*
width	*Width(s) in characters units* *Allowed: Integer* *Default: 20*
height	*Height in characters units* *Allowed: Integer* *Default: 6*
mode	*Listbox mode* *Allowed: String from 'browse', 'single', 'extended', 'multiple'* *Default: 'browse'*
font	*Font(s) of text* *Allowed: X font* *Default: ''*
relief	*Edge relief(s)* *Allowed: String from 'flat', 'ridge', 'raised', 'sunken', 'groove'* *Default: 'sunken'*
borderwidth	*Border width* *Allowed: Integer* *Default: 2*
foreground	*Color(s) of text* *Allowed: X color* *Default: 'black'*
background	*Background color(s)* *Allowed: X color* *Default: 'lightgrey'*
exportselection	*export to X clipboard?* *Allowed: Boolean* *Default: F*
fill	*how to expand when resized* *Allowed: String from 'x', 'y', 'both', 'none'* *Default: 'both'*
widgetset	*Widget set to use* *Allowed: A widgetserver closure tool* *Default: dws*

Returns
synclistboxes widget

Description
Constructs a synclistboxes according to the specified construction arguments, and returns a ref to the widget.
The arguments described with (s), e.g. width, may be either single values or multiple values. A single value argument applies to all of the listboxes. A multiple value argument must have the same number of elements as the number of listboxes in the widget and each argument is applied, in turn, to the listboxes in the order in which they are created.

---

synclistboxes.done - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Destroy ourselves

Synopsis
done()

Returns
Bool

Description

This function destroys the widget.

---

synclistboxes.listbox - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Return an underlying listbox agent

Synopsis
listbox(whichbox)

Arguments

whichbox	*Which listbox to return* *Allowed: Integer or unset* *Default: current leadbox*	

Returns
Tk listbox agent

Description

This function returns an underlying listbox agent. Be careful when using this as it is possible to delete the underlying listbox. Also, insertions made directly to the listbox will not trigger the automatic "see" of the end of the listbox which would normally occur when seeoninsert is set to T. It is also possible to get the listboxes out of sync using this function. The default listbox to return, if the whichbox argument is not supplied or is unset, is the listbox indicated by the leadbox argument.

---

synclistboxes.background - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Set the background color.

Synopsis
background(background, whichbox)

Arguments

background	*the new background color* *Allowed: X color*
whichbox	*which listbox gets the new color* *Allowed: An optional integer*

Description

This is an event which sets the background color one or all listboxes. If there is only one argument, all listboxes are set to that background color. If there is a second argument, it specifies which listbox should be set to that background color. Listboxes are numbered started at 1 up to the total number of listboxes. They are numbered in the order in which they were created.

Example

```
# slb is a previously created synclistboxes having 4 listboxes
# set the background color of all listboxes to lightgrey
slb->background('lightgrey');
# set the background color of listbox 3 to white
slb->background('white',3);
```

---

synclistboxes.bind - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

bind an event to a glish event

Synopsis

bind(xevent, gevent, whichbox)

Arguments

xevent	*the x event to be bound* *Allowed: X event*
gevent	*the glish event to be generated* *Allowed: String*
whichbox	*which listbox should this be bound to* *Allowed: An optional integer*

Description

This event works just like other Tk bind events in glish. The only difference is that there is an optional third argument which specifies which listbox this binding will apply to. If omitted, the binding applies to all listboxes in the widget. The widget *will emit an event having the name gevent whenever xevent occurs within the listboxes which the binding applies to.*

Example

```
# slb is a previously created synclistboxes having 4 listboxes
# bind mouse button 2 to all listboxes
slb->bind('<Button-2>','mb2');
# any button 2 events in any listbox will result in an "mb2" event
# being seen from slb
# bind mouse button 3 to the 2nd listbox
slb->bind('<Button-3>', 'mb3', 2);
# any button 3 events which occur within listbox 2 will result in an
# "mb3" event being seen from slb.
```

---

synclistboxes.borderwidth - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

change the borderwidth(s)

Synopsis
borderwidth(borderwidth, whichbox)

Arguments

borderwidth	*the new borderwidth* *Allowed: Integer*
whichbox	*which listbox gets the new borderwidth* *Allowed: An optional integer*

Description
This is an event which sets the borderwidth for one or all listboxes. If there is only one argument, all listboxes are set to that borderwidth. If there is a second argument, it specifies which listbox should be set to that borderwidth.

Example

```
# slb is a previously created synclistboxes having 4 listboxes
# set the borderwidth of lb 2 to be 3
slb->borderwidth(3,2);
```

---

synclistboxes.clear - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

clear the selection at the indicated items

Synopsis
clear(start, end)

Arguments

start	*the start of the range to clear* *Allowed: string*
end	*optional end of the range to clear* *Allowed: string*

Description
This event is just like the normal listbox clear event. It is applied to all listboxes within the widget.

Example

```
# slb is a previously created synclistboxes
# clear all selections
slb->clear('start','end')
```

---

synclistboxes.delete - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

delete the indicated items

Synopsis
delete(start, end)

Arguments

start	*the start of the range to delete* *Allowed: string*
end	*optional end of the range to delete* *Allowed: string*

Description
This event is just like the normal listbox delete event. It is applied to all listboxes within the widget.

Example

```
# slb is a previously created synclistboxes
# delete everything
slb->delete('start','end')
```

---

synclistboxes.exportselection - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

export the selection to the X clipboard?

Synopsis
exportselection(torf)

Arguments

torf	*export the selection to the X clipboard?* *Allowed: Boolean*

Description

This event is just like the normal listbox exportselection event. It is applied to all listboxes within the widget.

---

synclistboxes.foreground - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Set the foreground color.

Synopsis

foreground(foreground, whichbox)

Arguments

foreground	*the new foreground color* *Allowed: X color*
whichbox	*which listbox gets the new color* *Allowed: An optional integer*

Description

This is an event which sets the foreground color in one or all listboxes. If there is only one argument, all listboxes are set to that foreground color. If there is a second argument, it specifies which listbox should be set to that foreground color. Listboxes are numbered started at 1 up to the total number of listboxes. They are numbered in the order in which they were created. See the description for the background event for a related example.

---

synclistboxes.font - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

change the text font

Synopsis

font(font, whichbox)

Arguments

font	*The new text font* *Allowed: X font*
whichbox	*which listbox gets the new font* *Allowed: An optional integer*

Description
This is an event which sets the text font in one or all listboxes. If there is only one argument, all listboxes are set to that text font. If there is a second argument, it specifies which listbox should be set to that text font.

---

synclistboxes.get - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

request items, i.e. map indexes to items

Synopsis
get(indexes)

Arguments

indexes	*the requested indexes* *Allowed: Integer(s)*

Returns
glish record of items

Description
All of the values in each of the listboxes at the requested indexes are returned by using this event. The returned value is a record which can be indexed by listbox number, i.e. if the return value is result, then result[1] contains the requested items for listbox 1, result[2] the items for listbox 2, etc.

Example

```
# slb is a previously created synclistboxes having 4 listboxes
# get the items at indexes [1,2,4]
g := slb->get([1,2,4]);
# g[1] has the values for listbox 1, g[2] for listbox 2, etc.
```

---

synclistboxes.height - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Set the height

Synopsis
height(height)

Arguments

height	*the new height* *Allowed: Integer*

Description
This event is just like the normal listbox height event. It is applied to all listboxes within the widget.

---

synclistboxes.hscrollbar - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Turn on or off the horizontal scrollbars

Synopsis
hscrollbar(torf)

Arguments

torf	*Show a horizontal scrollbar?* *Allowed: Boolean*

Description
This is an event which turns on or off all of the horizontal scrollbars. They must either be all on or all off to preserve syncronization.

Example

```
# slb is a previously created synclistboxes
# turn off the horizontal scrollbar
slb->hscrollbar(F);
# turn it back on
slb->hscrollbar(T);
```

---

synclistboxes.insert - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

insert values into the listboxes

Synopsis
insert(values)

Arguments

values	*insert these values* *Allowed: A vector or matrix of strings*

Description
In order to remain in sync, the listboxes must always each have the same number of items as the other listboxes. This is enforced by making the delete event apply to all listboxes and requiring that the insert event value have a integer multiple of nboxes elements. For a single vector there must be exactly nboxes elements. Each

element is then placed in each listbox in order. To place multiple elements in each listbox using a single insert event, the value of that event should have a shape equal to [nboxes, nelements]. In other words, to place 7 elements into each of 3 listboxes the insert value must have a shape of [3, 7].
Insertions are currently all done at the end of the listboxes.

Example

```
# slb is a previously created synclistboxes having 2 listboxes
# insert 9 elements.
somevals := array(' ',2,9);
somevals[1,] := "one two three four five six seven eight nine";
somevals[2,] := "red orange yellow lightgrey blue brown peach mauve gold";
slb->insert(somevals);
```

---

synclistboxes.leadbox - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

which listbox is the default returned by the listbox function

Synopsis
leadbox(leadbox)

Arguments

leadbox	*which listbox takes the lead* *Allowed: Integer*

Description
As explain under the listbox function, popuphelp requires access to a true widget to act as the lead widget for positioning popuphelp. It knows to look for a listbox() function to supply that widget. The leadbox value determines which listbox is to act as the default lead widget when listbox is called with no arguments.

synclistboxes.mode - Function

Package	**display**

Module	**widgets**
Tool	**synclistboxes**

change listbox mode

Synopsis
mode(mode)

Arguments

mode	*change listbox mode* *Allowed: String from 'browse', 'single', 'multiple', or 'extended'*

Description
This event is just like the normal listbox mode event. It is applied to all listboxes within the widget.

---

synclistboxes.nearest - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

return item nearest the y offset param

Synopsis
nearest(nearest)

Arguments

nearest	*y offset param* *Allowed: Integer*

Returns
integer

Description
This event is just like the normal listbox nearest event. It is applied to first listbox (all listboxes would return the same value assuming they are all in sync).

---

synclistboxes.relief - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

change the border relief

Synopsis
relief(relief, whichbox)

Arguments

relief	*The new border relief* *Allowed: Standard relief string*
whichbox	*which listbox gets the new relief* *Allowed: An optional integer*

Description
This is an event which sets the border relieft in one or all listboxes. If there is only one argument, all listboxes are set to that relief. If there is a second argument, it specifies which listbox should be set to that text relief.

---

synclistboxes.see - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

scroll listboxes to specified index is visible

Synopsis
see(index)

Arguments

index	*the index to scroll to* *Allowed: String*

Description
This event is just like the normal listbox see event. It is applied to all listboxes within the widget.

---

synclistboxes.seeoninsert - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

See the end after each insert?

Synopsis
seeoninsert(torf)

Arguments

torf	*See the end after each insert?* *Allowed: Boolean*

Description

This event turns on or off the action which this widget does after each insert event. When on (T) a "see" event is sent to the listboxes to show the end of the listboxes after each inset event. When off (T) no such "see" event is sent.

Example

```
# slb is a previously created synclistboxes
# don't see the end after each insert event
slb->seeoninsert(F);
# see the after each insert event
slb->seeoninsertT);
```

---

synclistboxes.select - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

select the indicate items

Synopsis
select(indexex)

Arguments

indexex	*the indexes to select* *Allowed: String*

Description
This event is just like the normal listbox select event. It is applied to all listboxes within the widget.
This widget also emits a select event whenever the user makes a selection graphically.

---

synclistboxes.selection - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

get indexes of the current selection

Synopsis
selection()

Returns
integer

Description
This event is just like the normal listbox selection event. It is applied to the first

listbox in the widget (all the listboxes should return the same result if they are still in sync).
This widget also emits a select event whenever the user makes a selection graphically.

---

synclistboxes.vscrollbar - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Turn on or off the vertical scrollbar

Synopsis
vscrollbar(torf)

Arguments

torf	*Show a vertical scrollbar?* *Allowed: Boolean*

Description
This is an event which turns on or off the verticall scrollbar. See the description of hscrollbar for a similar example.

---

synclistboxes.width - Function

Package	**display**
Module	**widgets**
Tool	**synclistboxes**

Set the width

Synopsis
width(width, whichbox)

Arguments