Arguments

newheight	*The new listbox height.* *Allowed: An integer.*

Description
This value is only important if there is a vertical scrollbar present. If the vertical scrollbar is always on, the listbox is always this height. If the vertical scrollbar is only on when needed, the listbox is this height when the number of entries in the combobox exceeds this number, otherwise the listbox is only as high as it needs to be to hold the number of items in the combobox. If there is no scrollbar, the listbox will be whatever height is necessary to see all of the combobox entries at the same time without scrolling.

1.4.8 deleteregions - Tool

Package display
Module widgets

Tk widget to delete image regions from a Table
include "deleteregions.g"

Constructors

deleteregions	*Construct a deleteregions widget*

Functions

done	*Destroy ourselves*
gui	*Restart the GUI*
refresh	*Refresh the display*
setregions	*Set new regions*
settable	*Set a (new) table*

Description

This Tk widget enables you to delete `regions-of-interest` *(or more simply, regions - see Regionmanager) stored in an AIPS++ Table. It can also be used to provide a GUI interface for deleting regions from a record or global name space (from* Glish*).*
The widget displays, in listboxes, the regions stored in the Table (or record or

Glish*). The ‘Delete’ button is used to delete the regions that you have selected. The ‘Delete all’ button deletes all regions. The ‘Deselect’ button deselects all regions. Optionally, you can be prompted for confirmation before deletion by selecting from the ‘Confirm’ menu. The ‘Dismiss’ button dismisses the GUI. You can recover it with the gui function.*
Under the ‘File’ menu, you can either dismiss the GUI or destroy the widget (same as invoking function done).
The Table can be specified in many ways (at construction and via the entry box in the GUI); a Table `tool`*, an Image)* `tool`*, the name of a Table (a string) which might be an AIPS++ image, or the name of a* Glish *symbol which is either a Table or an Image* `tool`*.*
This widget emits an event called `deleted` *(it has no value) when the regions have been deleted. It also emits an event called* `dismissed` *(it has no value) when the dismiss button on the GUI is pressed.*

deleteregions - Constructor

Package	**display**
Module	**widgets**
Tool	**deleteregions**

Construct a deleteregions widget

Synopsis
deleteregions(parent, table, regions, source, widgetset)

Arguments

parent	*Parent for the widget* *Allowed: Widget*
table	*The table of interest* *Allowed: Table tool, image tool or string* *Default: ''*
regions	*Regions or region names (for source 'record' or 'global'* *Allowed: Record or vector of strings* *Default: ""*
source	*Source of the regions* *Allowed: String from 'table', 'record', or 'global'* *Default: 'table'*
widgetset	*The widget server to use* *Allowed: widgetserver tool* *Default: dws*

Returns

Agent

Description

This is the constructor. If `parent=F`*, then the widget will make its own parent frame.*
The argument `source` *specifies what the source of the regions is.*
If `source='table'` *then the regions are to be found in the Table specified with the argument* `table`*. The argument* `regions` *is ignored in this case. If you leave* `table` *unspecified, then you can enter the Table into the entry box provided in the GUI.*
The Table can be specified in many ways (at construction and via the entry box in the GUI); a Table `tool`*, an Image)* `tool`*, the name of a Table (a string) which might be an AIPS++ image, or the name of a* Glish *symbol which is either a Table or an Image* `tool`*.*
If `source='global'` *then the regions are to be found in the global name space of* Glish*. The argument* `table` *is ignored in this case. You must specify the names of the regions as a vector of strings with the* `regions` *argument.*
If `source='record'` *then the regions are to be found in the record given by the argument* `regions`*. The argument* `table` *is ignored in this case.*

Example

```
- include 'deleteregions.g'
- include 'regionmanager.g'
- dr1 := deleteregions(table='hcn', source='table');
- whenever dr1->deleted do {
   print 'regions deleted from table';
  }
-
- r1 := drm.quarter();
- r2 := drm.box();
- r3 := drm.box([10,10],20,20]);
- dr2 := deleteregions(source='global', regions="r1 r2 r3");
- whenever dr2->deleted do {
   print 'regions deleted from Glish';
  }
-
- r1 := drm.quarter();
- r2 := drm.box();
- r3 := drm.box([10,10],[20,20]);
- rec := [=];
```

```
- rec['r1'] := r1; rec['r2'] := r2; rec['r3'] := r3;
- dr3 := deleteregions(source='record', regions=rec);
- whenever dr3->deleted do {
   print 'regions deleted from record';
  }
-
```

We show the three ways to source the regions.

---

deleteregions.done - Function

Package	**display**
Module	**widgets**
Tool	**deleteregions**

Destroy ourselves

Synopsis
done()

Returns
Bool

Description

This function destroys the widget.

Example

```
- r := deleteregions();
- r
ref [*agent*=<agent>, done=<function>, gui=<function>, settable=<function>]
- r.done();
T
- r
F
-
```

---

deleteregions.gui - Function

Package	**display**
Module	**widgets**
Tool	**deleteregions**

Restart the GUI

Synopsis
gui()

Returns
Bool

Description

When the widget is first constructed, it brings up a GUI. When you delete regions, the GUI is dismissed. This function restarts the GUI; it will have the same state as when it was dismissed.

---

deleteregions.refresh - Function

Package	**display**
Module	**widgets**
Tool	**deleteregions**

Refresh the display

Synopsis
refresh()

Returns
Bool

Description
If the source is from a Table, this function reaccesses the table to find the regions and redisplays them. Otherwise it just clears any selection.

---

deleteregions.setregions - Function

Package	**display**
Module	**widgets**
Tool	**deleteregions**

Set new regions

Synopsis
setregions(regions)

Arguments

regions	*Regions or region names (for source 'record' or 'global'* *Allowed: Record or vector of strings* *Default: ""*

Returns
Bool

Description

This function lets you set a new set of regions into the widget. The source of the regions is the same as at construction and must match.

Example

```
- include 'deleteregions.g'
- include 'regionmanager.g'
- r1 := drm.quarter();
- r2 := drm.box();
- r3 := drm.box([10,10],20,20]);
- rec := [=];
- rec[1] := r1; rec[2] := r2; rec[3] := r3;
- dr3 := deleteregions(source='record', regions=rec);
- rec := [=];
= rec[1] := r1; rec[2] := r2;
- dr3.setregions(rec);
T
```

---

deleteregions.settable - Function

Package display
Module widgets
Tool deleteregions

Set a (new) table

Synopsis
settable(table)

Arguments

table	*The table of interest* *Allowed: Table tool, image tool or string* *Default: ''*	

Returns
Bool or fail

Description

This function manually (i.e. not via the GUI interface) sets a (new) Table name. It is equivalent to doing it via the GUI. The Table can be specified in many ways (at construction and via the entry box in the GUI); a Table `tool`*, an Image)* `tool`*, the name of a Table (a string) which might be an AIPS++ image, or the name of a* Glish *symbol which is either a Table or an Image* `tool`*.*

Example

```
- include 'deleteregions.g'
- include 'image.g'
- include 'table.g'
- r := deleteregions(table='hcn');
- im := image('hcn2');
- t := table('hcn2');
- r.settable(im);
- r.settable('hcn2');
- r.settable(t);
- r.settable('im');
- r.settable('t');
```