1.4.15 guientry - Tool

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

Tool for generating widgets for type-in of a GUI
include "guientry.g"

Constructors

guientry	*Make a guientry tool*

Functions

array	*Make a widget for an array entry*
boolean	*Make a widget for a boolean entry*
booleanarray	*Make a widget for a boolean array entry*
check	*Make a widget for a check entry*
choice	*Make a widget for a choice entry*
direction	*Make a widget for a direction entry*
done	*Destroy this tool*
doppler	*Make a widget for a doppler measure entry*
epoch	*Make a widget for a epoch entry*
file	*Make a widget for a file entry*
frequency	*Make a widget for a frequency measure entry*
list	*Make a widget for an extensible list of entry widget*
measurecodes	*Make a widget for entry of measures codes*
minmaxhist	*Constructs a widget for the entry of a range.*
position	*Make a widget for a position entry*
quantity	*Make a widget for a quantity entry*
radialvelocity	*Make a widget for a radialvelocity measure entry*
range	*Make a widget for a range entry*
record	*Make a widget for a Record entry*
region	*Make a widget for a region entry*
scalar	*Make a widget for a scalar entry*
scalarmeasure	*Make a widget for a scalar measure entry*
string	*Make a widget for a string entry*
twoentry	*Make a widget for a range entry, providing two entry boxes next to each other labelled 'min' and 'max'. This is used by minmaxhist, and is not intended for independant use.*
type	*Return the type of this tool*
untyped	*Make a widget for a untyped entry*

Description

This tool generates Tk widgets for the entry of various types of Glish variables.

This tool is used in the autogui. Note that guientry is served by the widgetserver and so that should be used to get access to a guientry.
The exact behavior of the generated widget depends upon the type of entry but in general:

- *The user may type in a value or a variable.*
- *A value can entered as a number in the entry field.*
- *A variable is entered by name only.*
- *A value may also be* unset *by using the unset button under a popupmenu menu.*
- *Either original or default values may be restored using the buttons under the popupmenu menu.*
- *Values may also be inserted via the insert function. Either a value (*e.g. *0.7), a string (*e.g. *'0.7'), or a variable (*e.g. `x` *or* `sqrt(y)`*) may be entered.*
- *Once initialized, a given widget can be used only for one type of argument (*i.e. *integer, double, complex). An attempt to enter a value of another type will generate an error.*
- *The widget always returns a a variable of the appropriate type via the get function. If the value of the widget is illegal then an illegal value is returned. This may be check for using the* `is_illegal()` *function.*
- *Entry of a valid value by any means generates a value event that has as an argument the value entered.*
- *The widget can be constructed so that the user cannot enter values directly (*`editable=F`*). This is used in the autogui for output-only values.*

The various functions of guientry generate a subsequence that can be used to interact with the widget. The function and events supported by the subsequence are:

`.get()` *Function to get the current value. WYSIWYG applies.*

`.insert(entry)` *Function to insert a value into the widget.*

`->insert(entry)` *Event to insert a value into the widget.*

`->value(entry)` *Event to designate that a user entered a new value.*

`.insertandemit(entry)` *Function to insert a value into the widget and emit a value event if the value is valid.*

`.setwidth(width)` *Function to set the total width in characters of the entry field(s).*

`.clear()` *Function to clear the current value.*

`.addtowrench` *Function to add a callback function to the wrench.*

`hlp` *Argument to specify popup help*

Parsing of entries is performed using the entryparser tool.
Most widgets have a popup menu that allows any of a number of operations. The standard ones are:

Original *Revert the entry to the original value that was first displayed.*

Default *Revert the entry to the default value for this actual widget.*

Unset *Toggle unset value. If the current value is not unset then it is changed to be so, if it already is unset then the original value is set.*

Copy *Copy the current entry to the clipboard whence it may be retrieved using* `dcb.paste()`.

Paste *Paste the current entry from the clipboard. An entry can be inserted in the clipboard using* `dcb.copy(value)`.

In addition, other operations may be present, as appropiate. For example, those widgets for accessing files of various types can bring up an interactive file chooser. One may add functions to the wrench, using addtowrench. The signature of the function is as in the following example. Note the use of the function insertandemit to insert the value and emit a value event if the value is valid. This is obviously required if a value is to be inserted into the widget.

```
#
# Addition to the wrench
#
  private.buttons.antennas := private.guientry.array();
  private.getantennas := function(ref self, ref its) {
    include 'gopher.g';
    wider private;
    self.insertandemit(dgo.getantennas(private.ms.name()));
  }
  private.buttons.antennas.addtowrench('Interactive', private.getantennas);
```

Example

```
include 'guientry.g'
f := dws.frame();
ge := dws.guientry()
s := ge.scalar(f, value=30.0);
```

```
whenever s->value do {
  print 'Gui selected a scalar ', s.get();
}
q := ge.quantity(f, value='100Hz');
whenever q->value do {
  print 'Gui selected a quantity ', q.get();
}
m := ge.measure(f, value='dm.direction(\'b1950\', \'0deg\', \'0deg\')')
whenever q->value do {
  print 'Gui selected a measure ', m.get();
}
```

guientry - Constructor

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a guientry tool

Synopsis
guientry(width, relief, font, background, foreground, editablecolor, uneditablecolor, unsetcolor, illegalcolor, borderwidth, expand, widgetset)

Arguments

width	*Initial width of entry field (characters)* *Allowed: Integer* *Default: 30*
relief	*Edge relief* *Allowed: String from 'flat', 'ridge', 'raised', 'sunken', 'groove'* *Default: 'flat'*
font	*Font of text* *Allowed: X font* *Default: ''*
background	*Background color* *Allowed: X color* *Default: 'lightgrey'*
foreground	*Color of text* *Allowed: X color* *Default: 'black'*
editablecolor	*Color for editable values* *Allowed: X color* *Default: 'white'*
uneditablecolor	*Color for uneditabe values* *Allowed: X color* *Default: 'lightgray'*
unsetcolor	*Color for unset values* *Allowed: X color* *Default: 'yellow'*
illegalcolor	*Color for illegal values* *Allowed: X color* *Default: 'red'*
borderwidth	*Borderwidth* *Allowed: Integer* *Default: 1*
expand	*Expand policy* *Allowed: 'x'—'y'—'both'—'none'* *Default: 'none'*
widgetset	*Widget server to use* *Allowed: A widgetserver tool* *Default: dws*

Returns
guientry widget

Description
Constructs a guientry according to the specified construction arguments, and returns a ref to the tool.

Example

```
include 'guientry.g'
f := dws.frame();
ge := dws.guientry()
r := ge.scalar(f, value=30.0);
whenever r->value do {
  print 'Gui selected is ', r.get();
}
```

---

guientry.scalar - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a scalar entry

Synopsis

scalar(parent, value, default, options, allowunset, editable, hlp)

Description

Constructs a widget for the entry of scalar values. The widget can determine the type of the first entry (integer, double or complex) and ensure that the type doesn't change. If neither the value nor the default are set AND allowunset is false then the options argument can be used to set the value that will actually be used as the initial value.

The **allowed formats for user input** *are:*

- *A scalar (*e.g. `1.3245`*)*
- *The name of Glish variable or function that evaluates to a scalar.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: String or unset*
default	*Default value used to determine type of GUI* *Allowed: String*
options	*options* *Allowed: Glish value* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns

guientry widget

Example

```
include 'guientry.g'
f := dws.frame();
ge := dws.guientry()
sangle := ge.scalar(f, value=pi/2);
sfloat := ge.scalar(f, options=0.0);
sint := ge.scalar(f, options=0);
```

---

guientry.array - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for an array entry

Synopsis

array(parent, value, default, options, allowunset, editable, maxdisplay, hlp)

Description
Constructs a widget for the entry of array values.
The **allowed formats for user input** *are:*

- *A vector of numbers (*e.g. `[1.3, 4.5, 900.32]`*)*
- *The name of Glish variable or function that evaluates to a vector of numbers.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
maxdisplay	*Maximum number of elements that will be displayed* *Allowed: Positive integer* *Default: 100*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.booleanarray - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a boolean array entry

Synopsis
booleanarray(parent, value, default, options, allowunset, editable, maxdisplay, hlp)

Description
Constructs a widget for the entry of array values.
The **allowed formats for user input** *are:*

- *A vector of booleans (*e.g. `[T, F, F, T]`*)*
- *The name of Glish variable or function that evaluates to a vector of booleans.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: T or F* *Default: T*
default	*Default value* *Allowed: T or F* *Default: T*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
maxdisplay	*Maximum number of elements that will be displayed* *Allowed: Positive integer* *Default: 100*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.boolean - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a boolean entry

Synopsis
boolean(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of boolean values.
The **allowed formats for user input** *are:*

- *Interactively by selecting the desired value*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.choice - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a choice entry

Synopsis
choice(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the choice of one string from many.
The **allowed formats for user input** *are:*

- *Interactively by selecting the desired value*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*Possible choices* *Allowed: Vector of strings*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

Example

```
include 'guientry.g'
f := dws.frame();
ge := dws.guientry();
ch := ge.choice(f, value='a', options="a b c");
whenever ch->value do {
  print 'Choice selected is ', ch.get();
}
```

---

guientry.check - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a check entry

Synopsis
check(parent, value, default, options, allowunset, editable, nperline, hlp)

Description
Constructs a widget for the choice of many strings from many.

The **allowed formats for user input** *are:*

- *Interactively by selecting the desired values*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*Possible choices* *Allowed: Vector of strings*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
nperline	*Number of items per line* *Allowed: Positive integer* *Default: 4*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.file - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a file entry

Synopsis
file(parent, value, default, options, allowunset, editable, types, hlp)

Description
The **allowed format for user input** *is simply a file name, relative or absolute.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
types	*File types to show (catalog)* *Allowed: String*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

Description

Constructs a widget for the entry of files. The `types` *argument is the same as the types available in the Catalog GUI. They can be recovered by the Catalog function availabletypesutility:catalog.availabletypes.*

---

guientry.measurecodes - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for entry of measures codes

Synopsis
measurecodes(parent, value, default, options, type, allowunset, editable, hlp)

Description

Constructs a widget for the entry of measures reference codes. For each type of measure there is a corresponding list of reference codes. For example, `dm.listcodes(dm.frequency())` *gives the list for a 'frequency' measure.*
The **allowed formats for user input** *(i.e. for* `value` *and* `default`*) are:*

- *A measure code as drawn from the list of possible codes returned by the measures tool (see above).*
- *The name of Glish variable or function that evaluates to a measure code.*

Use `options` *to specify the measure type (see Measures) for the current known list (e.g. 'frequency', 'doppler', 'direction' etc).*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*type of measure: any valid measure* *Allowed: string* *Default: ''*
type	*Type of argument: unset means to evaluate* *Allowed: Variable* *Default: unset*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

guientry.scalarmeasure - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a scalar measure entry

Synopsis

scalarmeasure(parent, value, default, options, allowunset, editable, type, hlp)

Description

Constructs a widget for the entry of scalar measure values.
The **allowed formats for user input** *are:*

- *A measure as returned by the measures tool (*e.g. `dm.frequency('LSR', '10GHz')`*)*
- *The name of Glish variable or function that evaluates to a measure.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
type	*Type of argument: unset means to evaluate* *Allowed: Variable* *Default: unset*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.frequency - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a frequency measure entry

Synopsis
frequency(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of frequency measure value.
The **allowed formats for user input** *are:*

- *A measure as returned by the measures tool (*e.g. `dm.frequency('LSR', '10GHz')`*)*
- *The name of Glish variable or function that evaluates to a frequency measure.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.radialvelocity - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a radialvelocity measure entry

Synopsis
radialvelocity(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of radialvelocity measure value.
The **allowed formats for user input** *are:*

- *A measure as returned by the measures tool (*e.g. `dm.radialvelocity('LSR', '-100km/s')`*)*
- *The name of Glish variable or function that evaluates to a radialvelocity measure.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.doppler - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a doppler measure entry

Synopsis
doppler(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of doppler measure value.
The **allowed formats for user input** *are:*

- *A measure as returned by the measures tool (*e.g. `dm.doppler('RADIO', 0.001)`*)*
- *The name of Glish variable or function that evaluates to a doppler measure.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.position - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a position entry

Synopsis
position(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of measures position values.
The **allowed formats for user input** *are:*

- *The position reference frame may be chosen using the left hand button. The two formats are WGS84 and ITRF (World Geodetic System and International Terrestrial Reference Frame). The position quantity values should be either longitude (angle), latitude(angle) and height(length); or x,y,z (length). The allowed formats for lengths and angles are described in the quanta tool.*
- *An observatory location may be selected using the ObservatoryList button under the spanner/wrench.*
- *The name of Glish variable or function that evaluates to a position measure.*

Note that changing the reference frame **does not** *change the values in the entry boxes. Instead, it defines the frame with the currently specified values. If you wish to make a conversion, use the measures position gui.*
The options string can be used to specify vertically aligned entries.

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options 'vertical'—"* *Allowed: string* *Default: "*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.epoch - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a epoch entry

Synopsis
epoch(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of measures epoch values.
The **allowed formats for user input** *are:*

- *The current epoch may be selected using the Today button under the spanner/wrench.*
- *The name of Glish variable or function that evaluates to a measure.*

The default initial reference code is 'UTC'. This may be changed using the options argument.
Note that changing the reference frame **does not** *change the value in the entry box. Instead, it defines the frame with the currently specified value. If you wish to make a conversion, use the measures epoch gui.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*Initial reference code* *Allowed: string* *Default: 'UTC'*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns

guientry widget

---

guientry.direction - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a direction entry

Synopsis

direction(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of measures direction values.
The **allowed formats for user input** *are:*

- *The direction reference frame may be chosen using the left hand button, and the latitude and longitude entered in the next two entry boxes. The allowed formats for times and angles are described in the quanta tool.*
- *A planet may be selected using the left hand button*
- *One of a list of bright radio sources may be selected using the SourceList button under the spanner/wrench.*
- *The name of Glish variable or function that evaluates to a direction measure.*

Note that changing the reference frame **does not** *change the values in the entry boxes. Instead, it defines the frame with the currently specified values. If you wish to make a conversion, use the measures direction gui.*
The options string can be used to specify vertically aligned entries.

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options: 'vertical'—''* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.quantity - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a quantity entry

Synopsis

quantity(parent, value, default, type, options, allowunset, editable, hlp)

Description

Constructs a widget for the entry of quantity values.
The **allowed formats for user input** *are:*

- *A quantity as returned by the quanta tool (*e.g. `dq.quantity('10GHz')`*)*
- *The name of Glish variable or function that evaluates to a quantity.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
type	*Type of argument: unset means to evaluate* *Allowed: Variable* *Default: unset*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.range - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a range entry

Synopsis
range(parent, value, default, options, allowunset, editable, rmin, rmax, rresolution, provideentry, hlp)

Description
Constructs a widget for the entry of range values.

The **allowed formats for user input** *are:*

- *Interactively by moving the slider*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Value or unset* *Default: 0*
default	*Default value* *Allowed: Value* *Default: 0*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
rmin	*Minimum value* *Allowed: Number* *Default: 0.0*
rmax	*Maximum value* *Allowed: Number* *Default: 1.0*
rresolution	*Resolution of selection* *Allowed: Number* *Default: 0.1*
provideentry	*Include a type-in entry box as well* *Allowed: Bool* *Default: F*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.record - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a Record entry

Synopsis
record(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of record values.
The **allowed formats for user input** *are:*

- *A record (*e.g. `[height=640, width=480]`*)*
- *The name of Glish variable of function that evaluates to is a record.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.region - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a region entry

Synopsis
region(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of region values.
The **allowed formats for user input** *are:*

- *A region as returned by the regionmanager tool (*e.g. `drm.box()`*)*
- *The name of Glish variable or function that evaluates to a region.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.string - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a string entry

Synopsis
string(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of string values.
The **allowed formats for user input** *are:*

- *A string*
- *A set of strings, separated by newlines.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

---

guientry.untyped - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a untyped entry

Synopsis
untyped(parent, value, default, options, allowunset, editable, hlp)

Description
Constructs a widget for the entry of untyped values. The behavior requires some explanation:

- *In determining the type of a given input string or insert argument, various conversions are tried, and the first successful one retained.*
- *The search list of conversions can be set using the setsearch function and retrieved using the search function.*
- *The default search list is "measure array coordinates region record string untyped"*
- *Valid search list items are the entry types that the Entryparser* `tool` *recognizes.*

The **allowed formats for user input** *are:*

- *Any valid glish expression*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Variable or unset*
default	*Default value* *Allowed: Variable*
options	*options* *Allowed: string* *Default: "*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns
guientry widget

guientry.list - Function

Package **display**
Module **widgets**
Tool **guientry**

Make a widget for an extensible list of entry widget

Synopsis
list(parent, types, names, values, values, defaults, options, allowunset, editable, hlp)

Description
Constructs a widget for a list of guienty widgets. This allows one to construct a list of made up of elements each containing one or more widgets. One may insert and get the contents as usual.

*The type of each widget is specified in the types argument. The corresponding names are given in the names argument. These names determine the names of the fields in the record returned by get. The arguments for each widget are passed as records with the elements taken in sequence (*i.e. *the names are immaterial. allowunset and editable apply to all widgets.*

Arguments

parent	*Parent of the menu* *Allowed: Tk frame*
types	*Types of widgets* *Allowed: String of valid widget names*
names	*Name to be attached to each widget* *Allowed: Vector String*
values	*Values to insert on start* *Allowed: Record of variables or unset*
values	*Values to insert on start* *Allowed: Record of variables or unset*
defaults	*Default values* *Allowed: Record of Variables or unset*
options	*options* *Allowed: Record of vector strings* *Default: "*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
hlp	*Popup help string* *Allowed: String* *Default: Unset*

Returns

guientry widget

Example

```
s := dge.list(f, types="string direction",
  names="Source Direction", name='Source',
  defaults=unset, values=[Source='No name', Direction=dm.direction()],
  allowunset=T, editable=T);
print s.get();
[Source1=[Source=M87, Direction=[m0=[value=187.705929, unit=deg], m1=[value=12.3
91125, unit=deg], type=direction, refer=J2000]], Source2=[Source=OJ287, Directio
n=[m0=[value=133.703646, unit=deg], m1=[value=20.1085119, unit=deg], type=direct
ion, refer=J2000]], Source3=[Source=3C273B, Direction=[m0=[value=187.277917, uni
```

---

guientry.done - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Destroy this tool

Synopsis
done()

Returns
T

Description
Destroys this Guientry tool.

---

guientry.type - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Return the type of this tool

Synopsis
type()

Returns
String

Description
Returns 'guientry'

---

guientry.twoentry - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Make a widget for a range entry, providing two entry boxes next to each other labelled 'min' and 'max'. This is used by minmaxhist, and is not intended for independant use.

Synopsis

twoentry(parent, value, default, options, allowunset, editable, min, max, widths, hlp)

Description

Constructs a widget for the entry of a range.

The **allowed formats for user input** *are:*

- *The user enters numbers into the two entry boxes.*

Arguments

parent	*Parent of the widget* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Array* *Default: unset*
default	*Default value* *Allowed: Array* *Default: 0*
options	*options* *Allowed: string* *Default: ''*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
min	*Minimum value* *Allowed: Number* *Default: -1.0*
max	*Maximum value* *Allowed: Number* *Default: 1.0*
widths	*Width of the two boxes* *Allowed: Number* *Default: 0.1*
hlp	*Help message* *Allowed: string* *Default: F*

Returns
guientry widget

---

guientry.minmaxhist - Function

Package	**display**
Module	**widgets**
Tool	**guientry**

Constructs a widget for the entry of a range.

Synopsis

minmaxhist(parent, value, default, allowunset, editable, minvalue, maxvalue, hlp, histarray, imageunits, options)

Description

Make a widget for selection of a range with the ability to allow the user to choose the range from a histogram (using histogramgui). The histogram is displayed and the user can drag, move and resize a region corresponding to the data min and max.

The **allowed formats for user input** *are:*

- *The user enters numbers into the two entry boxes. The user can open the histogram window, and select min and max using the mouse.*

Arguments

parent	*Parent of the widget* *Allowed: Tk frame*
value	*Value to insert on start* *Allowed: Array* *Default: unset*
default	*Default value* *Allowed: Array* *Default: 0*
allowunset	*Allow unset values* *Allowed: Boolean* *Default: T*
editable	*Is the entry editable by the user* *Allowed: Boolean* *Default: T*
minvalue	*Minimum value* *Allowed: Number* *Default: -1.0*
maxvalue	*Maximum value* *Allowed: Number* *Default: 1.0*
hlp	*Help message* *Allowed: string* *Default: ''*
histarray	*Record containing info to use for histogram (in .values and .counts)* *Allowed: Record* *Default: unset*
imageunits	*Brightness units of image (for labels)* *Allowed: string* *Default: unset*
options	*Options* *Allowed: string* *Default: ''*

Returns
T

1.4.16 guientrytest - Function

Package display
Module widgets