uicontrol

Create user interface control object

Syntax

```
handle = uicontrol('PropertyName',PropertyValue,...)
handle = uicontrol(parent,'PropertyName',PropertyValue,...)
handle = uicontrol
uicontrol(uich)
```

Description

`uicontrol` creates a uicontrol graphics objects (user interface controls), which you use to implement graphical user interfaces.

`handle = uicontrol('PropertyName',PropertyValue,...)` creates a uicontrol and assigns the specified properties and values to it. It assigns the default values to any properties you do not specify. The default uicontrol style is a pushbutton. The default parent is the current figure. See Properties for information about these and other properties.

`handle = uicontrol(parent,'PropertyName',PropertyValue,...)` creates a uicontrol in the object specified by the handle, `parent`. If you also specify a different value for the `Parent` property, the value of the `Parent` property takes precedence. `parent` can be the handle of a figure, uipanel, or uibuttongroup.

`handle = uicontrol` creates a pushbutton in the current figure. The `uicontrol` function assigns all properties their default values.

`uicontrol(uich)` gives focus to the uicontrol specified by the handle, `uich`.

When selected, most uicontrol objects perform a predefined action. MATLAB supports numerous styles of uicontrols, each suited for a different purpose:

- Check boxes
- Editable text fields
- Frames
- List boxes
- Pop−up menus
- Push buttons
- Radio buttons
- Sliders
- Static text labels
- Toggle buttons

For information on using these uicontrols within GUIDE, the MATLAB GUI development environment, see

- Setting Component Properties −− the Property Inspector
- Programming Callbacks for GUI Components

Specifying the Uicontrol Style

To create a specific type of uicontrol, set the `Style` property as one of the following strings:

- `'checkbox'` − Check boxes generate an action when selected. These devices are useful when providing the user with a number of independent choices. To activate a check box, click the mouse button on the object. The state of the device is indicated on the display.
- `'edit'` − Editable text fields enable users to enter or modify text values. Use editable text when you want text as input. If `Max-Min>1`, then multiple lines are allowed. For multi−line edit boxes, a vertical scrollbar enables scrolling, as do the arrow keys.
- `'frame'` − Frames are rectangles that provide a visual enclosure for regions of a figure window. Frames can make a user interface easier to understand by grouping related controls. Frames have no callback routines associated with them. Only other uicontrols can appear within frames.

  Frames are opaque, not transparent, so the order in which you define uicontrols is important in determining whether uicontrols within a frame are covered by the frame or are visible. *Stacking order* determines the order objects are drawn: objects defined first are drawn first; objects defined later are drawn over existing objects. If you use a frame to enclose objects, you must define the frame before you define the objects.

  > **Note** Most frames in existing GUIs can now be replaced with panels (`uipanel`) or button groups (`uibuttongroup`). GUIDE continues to support frames in those GUIs that contain them, but the frame component does not appear in the GUIDE Layout Editor component palette.

- `'listbox'` − List boxes display a list of items (defined using the `String` property) and enable users to select one or more items. The `Min` and `Max` properties control the selection mode:

  If `Max-Min>1`, then multiple selection is allowed.

  If `Max-Min<=1`, then only single selection is allowed.

  The `Value` property indicates selected entries and contains the indices into the list of strings; a vector value indicates multiple selections. MATLAB evaluates the list box's callback routine after any mouse button up event that changes the `Value` property. Therefore, you may need to add a "Done" button to delay action caused by multiple clicks on list items. List boxes differentiate between single and double clicks and set the figure `SelectionType` property to `normal` or `open` accordingly before evaluating the list box's `Callback` property.

- `'popupmenu'` − Pop−up menus open to display a list of choices (defined using the `String` property) when pressed. When not open, a pop−up menu indicates the current choice. Pop−up menus are useful when you want to provide users with a number of mutually exclusive choices, but

do not want to take up the amount of space that a series of radio buttons requires. You must specify a value for the `String` property.

- `'pushbutton'` − Push buttons generate an action when pressed. To activate a push button, click the mouse button on the push button.
- `'radiobutton'` − Radio buttons are similar to check boxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one is in a pressed state at any given time). To activate a radio button, click the mouse button on the object. The state of the device is indicated on the display. Note that your code can implement the mutually exclusive behavior of radio buttons.
- `'slider'` − Sliders accept numeric input within a specific range by enabling the user to move a sliding bar. Users move the bar by pressing the mouse button and dragging the pointer over the bar, or by clicking in the trough or on an arrow. The location of the bar indicates a numeric value, which is selected by releasing the mouse button. You can set the minimum, maximum, and current values of the slider.
- `'text'` − Static text boxes display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively and there is no way to invoke the callback routine associated with it.
- `'togglebutton'` − Toggle buttons are controls that execute callbacks when clicked on and indicate their state, either on or off. Toggle buttons are useful for building toolbars.

Remarks

- The `uicontrol` function accepts property name/property value pairs, structures, and cell arrays as input arguments and optionally returns the handle of the created object. You can also set and query property values after creating the object using the `set` and `get` functions.
- A `uicontrol` object is a child of a figure, uipanel, or uibuttongroup and therefore does not require an axes to exist when placed in a figure window, uipanel, or uibuttongroup.
- When MATLAB is paused and a uicontrol has focus, pressing a keyboard key does not cause MATLAB to resume. Click anywhere outside a uicontrol and then press any key. See the `pause` function for more information.

Properties

This table lists all properties useful for `uicontrol` objects, grouping them by function. Each property name acts as a link to a description of the property.

Property Name	Property Description	Property Value
Controlling Style and Appearance		
`BackgroundColor`	Object background color	Value: `ColorSpec` Default: system dependent

CData	Truecolor image displayed on the control	Value: matrix
ForegroundColor	Color of text	Value: ColorSpec Default: [0 0 0]
SelectionHighlight	Object highlighted when selected	Value: on, off Default: on
String	Uicontrol label, also list box and pop−up menu items	Value: string
Visible	Uicontrol visibility	Value: on, off Default: on
General Information About the Object		
Children	Uicontrol objects have no children	
Enable	Enable or disable the uicontrol	Value: on, inactive, off Default: on
Parent	Uicontrol object's parent	Value: figure, uipanel, or uibuttongroup handle
Selected	Whether object is selected	Value: on, off Default: off
SliderStep	Slider step size	Value: two−element vector Default: [0.01 0.1]
Style	Type of uicontrol object	Value: pushbutton, togglebutton, radiobutton, checkbox, edit, text, slider, listbox, popupmenu Default: pushbutton
Tag	User−specified object identifier	Value: string
TooltipString	Content of object's tooltip	Value: string
Type	Class of graphics object	Value: string (read−only) Default: uicontrol
UserData	User−specified data	Value: matrix
Controlling the Object Position		
Position	Size and location of uicontrol object	Value: position rectangle Default: [20 20 60 20]

Units	Units to interpret position vector	Value: pixels, normalized, inches, centimeters, points, characters Default: pixels
Controlling Fonts and Labels		
FontAngle	Character slant	Value: normal, italic, oblique Default: normal
FontName	Font family	Value: string Default: system dependent
FontSize	Font size	Value: size in FontUnits Default: system dependent
FontUnits	Font size units	Value: points, normalized, inches, centimeters, pixels Default: points
FontWeight	Weight of text characters	Value: light, normal, demi, bold Default: normal
HorizontalAlignment	Alignment of label string	Value: left, center, right Default: depends on uicontrol object
String	Uicontrol object label, also list box and pop−up menu items	Value: string
Controlling Callback Routine Execution		
BusyAction	Callback routine interruption	Value: cancel, queue Default: queue
ButtonDownFcn	Button−press callback routine	Value: string or function handle
Callback	Control action	Value: string or function handle
CreateFcn	Callback routine executed during object creation	Value: string or function handle
DeleteFcn	Callback routine executed during object deletion	Value: string or function handle
Interruptible	Callback routine interruption mode	Value: on, off Default: on
KeyPressFcn	Key press callback routine	Value: string or function handle

UIContextMenu	Uicontextmenu object associated with the uicontrol	Value: handle
Information About the Current State		
ListboxTop	Index of top−most string displayed in list box	Value: scalar Default: [1]
Max	Maximum value (depends on uicontrol object)	Value: scalar Default: object dependent
Min	Minimum value (depends on uicontrol object)	Value: scalar Default: object dependent
Value	Current value of uicontrol object	Value: scalar or vector Default: object dependent
Controlling Access to Objects		
HandleVisibility	Whether handle is accessible from command line and GUIs	Value: on, callback, off Default: on
HitTest	Whether selectable by mouse click	Value: on, off Default: on

Examples

Example 1. The following statement creates a push button that clears the current axes when pressed.

```
h = uicontrol('Style', 'pushbutton', 'String', 'Clear',...
    'Position', [20 150 100 70], 'Callback', 'cla');
```

This statement gives focus to the pushbutton.

```
uicontrol(h)
```

Example 2. You can create a uicontrol object that changes figure colormaps by specifying a pop−up menu and supplying an M−file name as the object's `Callback`:

```
hpop = uicontrol('Style', 'popup',...
       'String', 'hsv|hot|cool|gray',...
       'Position', [20 320 100 50],...
       'Callback', 'setmap');
```

The above call to `uicontrol` defines four individual choices in the menu: `hsv`, `hot`, `cool`, and `gray`. You specify these choices with the `String` property, separating the choices with the "|" character.

The `Callback`, in this case `setmap`, is the name of an M−file that defines a more

complicated set of instructions than a single MATLAB command. setmap contains these statements:

```
val = get(hpop,'Value');
if val == 1
    colormap(hsv)
elseif val == 2
    colormap(hot)
elseif val == 3
    colormap(cool)
elseif val == 4
    colormap(gray)
end
```

The `Value` property contains a number that indicates the selected choice. The choices are numbered sequentially from one to four. The `setmap` M−file can get and then test the contents of the `Value` property to determine what action to take.

See Also

`textwrap`, `uibuttongroup`, `uimenu`, `uipanel`

Uicontextmenu Properties Uicontrol Properties