



Uicontrol Properties

Modifying Properties

You can set and query graphics object properties in two ways:

- The [Property Inspector](#) is an interactive tool that enables you to see and change object property values. The Property inspector is available from GUIDE, or use the [inspect](#) function at the command line.
- The [set](#) and [get](#) commands enable you to set and query the values of properties

To change the default value of properties see [Setting Default Property Values](#). You can also set default [uicontrol](#) properties on the root and figure levels:

```
set(0, 'DefaultUicontrolProperty', PropertyValue...)
set(gcf, 'DefaultUicontrolProperty', PropertyValue...)
```

where *Property* is the name of the [uicontrol](#) property whose default value you want to set and *PropertyValue* is the value you are specifying as the default. Use [set](#) and [get](#) to access [uicontrol](#) properties.

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](#)

Uicontrol Properties

This section lists all properties useful to [uicontrol](#) objects along with valid values and descriptions of their use. Curly braces {} enclose default values.

Property	Purpose
BackgroundColor	Object background color
BusyAction	Callback routine interruption
ButtonDownFcn	Button-press callback routine
Callback	Control action
CData	Truecolor image displayed on the control
Children	Uicontrol objects have no children
CreateFcn	Callback routine executed during object creation
DeleteFcn	Callback routine executed during object deletion
Enable	Enable or disable the uicontrol
FontAngle	Character slant
FontName	Font family

FontSize	Font size
FontUnits	Font size units
FontWeight	Weight of text characters
ForegroundColor	Color of text
HandleVisibility	Whether handle is accessible from command line and GUIs
HitTest	Whether selectable by mouse click
HorizontalAlignment	Alignment of label string
Interruptible	Callback routine interruption mode
KeyPressFcn	Key press callback routine
ListboxTop	Index of top–most string displayed in list box
Max	Maximum value (depends on uicontrol object)
Min	Minimum value (depends on uicontrol object)
Parent	Uicontrol object's parent
Position	Size and location of uicontrol object
Selected	Whether object is selected
SelectionHighlight	Object highlighted when selected
SliderStep	Slider step size
String	Uicontrol object label, also list box and pop–up menu items
Style	Type of uicontrol object
Tag	User–specified object identifier
TooltipString	Content of object's tooltip
Type	Class of graphics object
UIContextMenu	Uicontextmenu object associated with the uicontrol
Units	Units to interpret <code>position</code> vector
UserData	User–specified data
Value	Current value of uicontrol object
Visible	Uicontrol visibility

BackgroundColor

ColorSpec

Object background color. The color used to fill the **uicontrol** rectangle. Specify a color using a three–element RGB vector or one of the MATLAB predefined names. The default color is determined by system settings. See [ColorSpec](#) for more information on specifying color.

BusyAction cancel | {queue}

Callback routine interruption. If a callback is executing and the user triggers an event (such as a mouse click) on an object for which a callback is defined, the callback associated with the new event uses the value of `BusyAction` to decide whether or not to attempt to interrupt the executing callback.

- If the value is `cancel`, the event is discarded and the second callback does not execute.
- If the value is `queue`, and the `Interruptible` property of the first callback is `on`, the second callback is added to the event queue and executes in its turn after the first callback finishes execution.

Note If the interrupting callback is a `DeleteFcn` or `CreateFcn` callback or a figure's `CloseRequest` or `ResizeFcn` callback, it interrupts an executing callback regardless of the value of that object's `Interruptible` property. See the [Interruptible](#) property for information about controlling a callback's interruptibility.

ButtonDownFcn string or function handle (GUIDE sets this property)

Button-press callback routine. A callback routine that can execute when you press a mouse button while the pointer is on or near `uicontrol`. Specifically:

- If the `uicontrol`'s `Enable` property is set to `on`, the `ButtonDownFcn` callback executes when you click the right or left mouse button in a 5-pixel border around the `uicontrol` or when you click the right mouse button on the control itself.
- If the `uicontrol`'s `Enable` property is set to `inactive` or `off`, the `ButtonDownFcn` executes when you click the right or left mouse button in the 5-pixel border or on the control itself.

This is useful for implementing actions to interactively modify control object properties, such as size and position, when they are clicked on (using `selectmoveresize`, for example).

Define this routine as a string that is a valid MATLAB expression or the name of an M-file. The expression executes in the MATLAB workspace.

To add a `ButtonDownFcn` callback in GUIDE, select `View Callbacks` from the Layout Editor **View** menu, then select `ButtonDownFcn`. GUIDE sets this property to the appropriate string and adds the callback to the M-file the next time you save the GUI. Alternatively, you can set this property to the string `%automatic`. The next time you save the GUI, GUIDE sets this property to the appropriate string and adds the callback to the M-file.

Use the `Callback` property to specify the callback routine that executes when you activate the enabled `uicontrol` (e.g., click on a push button).

Callback string (GUIDE sets this property)

Control action. A routine that executes whenever you activate the `uicontrol` object (e.g., when you click on a push button or move a slider). Define this

routine as a string that is a valid MATLAB expression or the name of an M-file. The expression executes in the MATLAB workspace.

To execute the callback routine for an edit text control, type in the desired text and then do one of the following:

- Click another component, the menu bar, or the background of the GUI.
- For a single line editable text box, press Enter, or
- For a multiline editable text box, press Ctrl+Enter.

Callback routines defined for static text do not execute because no action is associated with these objects.

CData matrix

Tricolor image displayed on control A three-dimensional matrix of RGB values that defines a tricolor image displayed on either a push button or toggle button. Each value must be between 0.0 and 1.0.

Children matrix

The empty matrix; **uicontrol** objects have no children.

Clipping {on} | off

This property has no effect on uicontrols.

CreateFcn string

Callback routine executed during object creation The specified function executes when MATLAB creates a **uicontrol** object. MATLAB sets all property values for the uicontrol before executing the **CreateFcn** callback so these values are available to the callback. Within the function, use `gcbob` to get the handle of the **uicontrol** being created.

Setting this property on an existing **uicontrol** object has no effect.

You can define a default **CreateFcn** callback for all new uicontrols. This default applies unless you override it by specifying a different **CreateFcn** callback when you call **uicontrol**. For example, the code

```
set(0,'DefaultUicontrolCreateFcn','set(gcf,'BackgroundColor','white'))
```

creates a default **CreateFcn** callback that runs whenever you create a new **uicontrol**. It sets the default background color of all new uicontrols.

To override this default and create a **uicontrol** whose **BackgroundColor** is set to a different value, call **uicontrol** with code similar to

```
hpt = uicontrol(...,'CreateFcn','set(gcf,'BackgroundColor','blue'))
```

Note To override a default `CreateFcn` callback you must provide a new callback and not just provide different values for the specified properties. This is because the `CreateFcn` callback runs after the property values are set, and can override property values you have set explicitly in the `uicontrol` call. In the example above, if instead of redefining the `CreateFcn` property for this `uicontrol`, you had explicitly set `BackgroundColor` to `blue`, the default `CreateFcn` callback would have set `BackgroundColor` back to the default, i.e., `white`.

See [Function Handle Callbacks](#) for information on how to use function handles to define a callback function.

DeleteFcn string

Delete `uicontrol` callback routine. A callback routine that executes when you delete the `uicontrol` object (e.g., when you issue a `delete` command or clear the figure containing the `uicontrol`). MATLAB executes the routine before destroying the object's properties so these values are available to the callback routine.

The handle of the object whose `DeleteFcn` is being executed is accessible only through the root `CallbackObject` property, which you can query using `gcbo`.

See [Function Handle Callbacks](#) for information on how to use function handles to define a callback function.

Enable {on} | inactive | off

Enable or disable the `uicontrol`. This property controls how `uicontrols` respond to mouse button clicks, including which callback routines execute.

- `on` – The `uicontrol` is operational (the default).
- `inactive` – The `uicontrol` is not operational, but looks the same as when `Enable` is `on`.
- `off` – The `uicontrol` is not operational and its image (set by the `Cdata` property) is grayed out.

When you left-click on a `uicontrol` whose `Enable` property is `on`, MATLAB performs these actions in this order:

1. Sets the figure's `SelectionType` property.
2. Executes the `uicontrol`'s `ClickedCallback` routine.
3. Does not set the figure's `CurrentPoint` property and does not execute either the control's `ButtonDownFcn` or the figure's `WindowButtonDownFcn` callback.

When you left-click on a `uicontrol` whose `Enable` property is `off`, or when you right-click a `uicontrol` whose `Enable` property has any value, MATLAB performs these actions in this order:

1. Sets the figure's `SelectionType` property.
2. Sets the figure's `CurrentPoint` property.
3. Executes the figure's `WindowButtonDownFcn` callback.

Extent position rectangle (read only)

*Size of **uicontrol** character string.* A four–element vector that defines the size and position of the character string used to label the **uicontrol**. It has the form:

```
[0,0,width,height]
```

The first two elements are always zero. `width` and `height` are the dimensions of the rectangle. All measurements are in units specified by the `Units` property.

Since the `Extent` property is defined in the same units as the **uicontrol** itself, you can use this property to determine proper sizing for the **uicontrol** with regard to its label. Do this by

- Defining the `String` property and selecting the font using the relevant properties.
- Getting the value of the `Extent` property.
- Defining the `width` and `height` of the `Position` property to be somewhat larger than the `width` and `height` of the `Extent`.

For multiline strings, the `Extent` rectangle encompasses all the lines of text. For single line strings, the `Extent` is returned as a single line, even if the string wraps when displayed on the control.

FontAngle {normal} | italic | oblique

Character slant. MATLAB uses this property to select a font from those available on your particular system. Setting this property to `italic` or `oblique` selects a slanted version of the font, when it is available on your system.

FontName string

Font family. The name of the font in which to display the `String`. To display and print properly, this must be a font that your system supports. The default font is system dependent.

To use a fixed–width font that looks good in any locale (and displays properly in Japan, where multibyte character sets are used), set `FontName` to the string `FixedWidth` (this string value is case sensitive):

```
set(uicontrol_handle, 'FontName', 'FixedWidth')
```

This parameter value eliminates the need to hard code the name of a fixed–width font, which may not display text properly on systems that do not use ASCII character encoding (such as in Japan). A properly written MATLAB application that needs to use a fixed–width font should set `FontName` to `FixedWidth` and rely on the `rootFixedWidthFontName` property to be set correctly in the end user’s environment.

End users can adapt a MATLAB application to different locales or personal environments by setting the `rootFixedWidthFontName` property to the appropriate value for that locale from `startup.m`. Setting the `rootFixedWidthFontName` property causes an immediate update of the display to use the new font.

FontSize size in `FontUnits`

Font size. A number specifying the size of the font in which to display the

String, in units determined by the `FontUnits` property. The default point size is system dependent.

FontUnits {points} | normalized | inches | centimeters | pixels

Font size units. This property determines the units used by the `FontSize` property. Normalized units interpret `FontSize` as a fraction of the height of the `uicontrol`. When you resize the `uicontrol`, MATLAB modifies the screen `FontSize` accordingly. `pixels`, `inches`, `centimeters`, and `points` are absolute units (1 point = 1/72 inch).

FontWeight light | {normal} | demi | bold

Weight of text characters. MATLAB uses this property to select a font from those available on your particular system. Setting this property to `bold` causes MATLAB to use a bold version of the font, when it is available on your system.

ForegroundColor [ColorSpec](#)

Color of text. This property determines the color of the text defined for the `String` property (the `uicontrol` label). Specify a color using a three-element RGB vector or one of MATLAB's predefined names. The default text color is black. See [ColorSpec](#) for more information on specifying color.

HandleVisibility {on} | callback | off

Control access to object's handle. This property determines when an object's handle is visible in its parent's list of children. When a handle is not visible in its parent's list of children, it is not returned by functions that obtain handles by searching the object hierarchy or querying handle properties. This includes [set](#), [findobj](#), [gca](#), [gcf](#), [gco](#), [newplot](#), [cla](#), [clf](#), and [close](#). Neither is the handle visible in the parent figure's `CurrentObject` property. Handles that are hidden are still valid. If you know an object's handle, you can [set](#) and [get](#) its properties, and pass it to any function that operates on handles.

- Handles are always visible when `HandleVisibility` is `on`.
- Setting `HandleVisibility` to `callback` causes handles to be visible from within callback routines or functions invoked by callback routines, but not from within functions invoked from the command line. This provides a means to protect GUIs from command-line users, while allowing callback routines to have complete access to object handles.
- Setting `HandleVisibility` to `off` makes handles invisible at all times. This may be necessary when a callback routine invokes a function that might potentially damage the GUI (such as evaluating a user-typed string), and so temporarily hides its own handles during the execution of that function.

You can set the root [ShowHiddenHandles](#) property to `on` to make all handles visible, regardless of their `HandleVisibility` settings. This does not affect the values of the `HandleVisibility` properties.

Note Radio buttons and toggle buttons that are managed by a `uibuttongroup` should not be accessed outside the button group. Set the `HandleVisibility` of such radio buttons and toggle buttons to `off` to prevent inadvertent access.

HitTest {on} | off

Selectable by mouse click. This property has no effect on `uicontrol` objects.

HorizontalAlignment left | {center} | right

Horizontal alignment of label string This property determines the justification of the text defined for the `String` property (the `uicontrol` label):

- `left` -- Text is left justified with respect to the `uicontrol`.
- `center` -- Text is centered with respect to the `uicontrol`.
- `right` -- Text is right justified with respect to the `uicontrol`.

On Microsoft Windows systems, this property affects only `edit` and `text` `uicontrols`.

Interruptible {on} | off

Callback routine interruption mode If a callback is executing and the user triggers an event (such as a mouse click) on an object for which a callback is defined, that callback attempts to interrupt the first callback. MATLAB processes the callbacks according to these factors:

- The `Interruptible` property of the object whose callback is executing
- Whether the executing callback contains `drawnow`, `figure`, `getframe`, `pause`, or `waitfor` statements
- The `BusyAction` property of the object whose callback is waiting to execute

If the `Interruptible` property of the object whose callback is executing is `on` (the default), the callback can be interrupted. Whenever the callback calls one of the `drawnow`, `figure`, `getframe`, `pause`, or `waitfor` functions, the function processes any events in the event queue, including the waiting callback, before performing its defined task.

If the `Interruptible` property of the object whose callback is executing is `off`, the callback cannot be interrupted (except by certain callbacks; see the note below). The `BusyAction` property of the object whose callback is waiting to execute determines what happens to the callback.

Note If the interrupting callback is a `DeleteFcn` or `CreateFcn` callback or a figure's `CloseRequest` or `ResizeFcn` callback, it interrupts an executing callback regardless of the value of that object's `Interruptible` property. The interrupting callback starts execution at the next `drawnow`, `figure`, `getframe`, `pause`, or `waitfor` statement. A figure's `WindowButtonDownFcn` callback routine, or an object's `ButtonDownFcn` or `Callback` routine are processed according to the rules described above.

KeyPressFcn

string or function handle

Key press callback function. A callback routine invoked by a key press when the callback's **uicontrol** object has focus. Focus is denoted by a border or a dotted border, respectively, in UNIX and Microsoft Windows. If no **uicontrol** has focus, the figure's key press callback function, if any, is invoked. `KeyPressFcn` can be a function handle, the name of an M-file, or any legal MATLAB expression.

If the specified value is the name of an M-file, the callback routine can query the figure's `CurrentCharacter` property to determine what particular key was pressed and thereby limit the callback execution to specific keys.

If the specified value is a function handle, the callback routine can retrieve information about the key that was pressed from its event data structure argument.

Event Data Structure Field	Description	Examples:			
		a	=	Shift	Shift/a
Character	Character interpretation of the key that was pressed.	'a'	'='	''	'A'
Modifier	Current modifier, such as 'control', or an empty cell array if there is no modifier	{1x0 cell}	{1x0 cell}	{'shift'}	{'shift'}
Key	Name of the key that was pressed.	'a'	'equal'	'shift'	'a'

See [Function Handle Callbacks](#) for information on how to use function handles to define the callback function.

ListboxTop

scalar

Index of top-most string displayed in list box. This property applies only to the `listbox` style of **uicontrol**. It specifies which string appears in the top-most position in a list box that is not large enough to display all list entries.

`ListboxTop` is an index into the array of strings defined by the `String` property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer.

Max

scalar

Maximum value. This property specifies the largest value allowed for the `Value` property. Different styles of `uicontrol`s interpret `Max` differently:

- Check boxes – `Max` is the setting of the `Value` property while the check box is selected.

- Editable text – If $Max - Min > 1$, then editable text boxes accept multiline input. If $Max - Min \leq 1$, then editable text boxes accept only single line input.
- List boxes – If $Max - Min > 1$, then list boxes allow multiple item selection. If $Max - Min \leq 1$, then list boxes do not allow multiple item selection.
- Radio buttons – Max is the setting of the `Value` property when the radio button is selected.
- Sliders – Max is the maximum slider value and must be greater than the Min property. The default is 1.
- Toggle buttons – Max is the value of the `Value` property when the toggle button is selected. The default is 1.
- Pop-up menus, push buttons, and static text do not use the Max property.

Min scalar

Minimum value. This property specifies the smallest value allowed for the `Value` property. Different styles of uicontrols interpret `Min` differently:

- Check boxes – Min is the setting of the `Value` property while the check box is not selected.
- Editable text – If $Max - Min > 1$, then editable text boxes accept multiline input. If $Max - Min \leq 1$, then editable text boxes accept only single line input.
- List boxes – If $Max - Min > 1$, then list boxes allow multiple item selection. If $Max - Min \leq 1$, then list boxes allow only single item selection.
- Radio buttons – Min is the setting of the `Value` property when the radio button is not selected.
- Sliders – Min is the minimum slider value and must be less than Max . The default is 0.
- Toggle buttons – Min is the value of the `Value` property when the toggle button is not selected. The default is 0.
- Pop-up menus, push buttons, and static text do not use the `Min` property.

Parent handle

Uicontrol parent. The handle of the `uicontrol`'s parent object. You can move a `uicontrol` object to another figure, `uipanel`, or `uibuttongroup` by setting this property to the handle of the new parent.

Position position rectangle

Size and location of uicontrol. The rectangle defined by this property specifies the size and location of the control within the parent figure window, `uipanel`, or `uibuttongroup`. Specify `Position` as

[left bottom width height]

`left` and `bottom` are the distance from the lower-left corner of the parent object to the lower-left corner of the `uicontrol` object. `width` and `height` are the dimensions of the `uicontrol` rectangle. All measurements are in units

specified by the `Units` property.

On Microsoft Windows systems, the height of pop-up menus is automatically determined by the size of the font. The value you specify for the `height` of the `Position` property has no effect.

The `width` and `height` values determine the orientation of sliders. If `width` is greater than `height`, then the slider is oriented horizontally. If `height` is greater than `width`, then the slider is oriented vertically.

Selected `on` | `{off}` (read only)

Is object selected. When this property is `on`, MATLAB displays selection handles if the `SelectionHighlight` property is also `on`. You can, for example, define the `ButtonDownFcn` to set this property, allowing users to select the object with the mouse.

SelectionHighlight `{on}` | `off`

Object highlight when selected When the `Selected` property is `on`, MATLAB indicates the selected state by drawing four edge handles and four corner handles. When `SelectionHighlight` is `off`, MATLAB does not draw the handles.

SliderStep `[min_step max_step]`

Slider step size. This property controls the amount the `sliderValue` changes when you click the mouse on the arrow button (`min_step`) or on the slider trough (`max_step`). Specify `SliderStep` as a two-element vector; each value must be in the range `[0, 1]`. The actual step size is a function of the specified `SliderStep` and the total slider range (`Max - Min`). The default, `[0.01 0.10]`, provides a 1 percent change for clicks on the arrow button and a 10 percent change for clicks in the trough.

For example, if you create the following slider,

```
uicontrol('Style','slider','Min',1,'Max',7,...  
         'SliderStep',[0.1 0.6])
```

clicking on the arrow button moves the indicator by,

```
0.1*(7-1)  
ans =  
    0.6000
```

and clicking in the trough moves the indicator by,

```
0.6*(7-1)  
ans =  
    3.6000
```

Note that if the specified step size moves the slider to a value outside the range, the indicator moves only to the `Max` or `Min` value.

See also the `Max`, `Min`, and `Value` properties.

string `string`

Uicontrol label, list box items, pop-up menu choices. For check boxes, editable text, push buttons, radio buttons, static text, and toggle buttons the text displayed on the object. For list boxes and pop-up menus, the set of entries or items displayed in the object.

For **uicontrol** objects that display only one line of text if the string value is specified as a cell array of strings or padded string matrix, only the first string of a cell array or of a padded string matrix is displayed; the rest are ignored. Vertical slash ('|') characters are not interpreted as line breaks and instead show up in the text displayed in the **uicontrol**.

For multiple line editable text or static text controls line breaks occur between each row of the string matrix, each cell of a cell array of strings, and after any \n characters embedded in the string. Vertical slash ('|') characters are not interpreted as line breaks, and instead show up in the text displayed in the **uicontrol**.

For multiple items on a list box or pop-up menu you can specify items as a cell array of strings, a padded string matrix, or within a string vector separated by vertical slash ('|') characters. Use the `Value` property to set the index of the initial item selected.

For check boxes, push buttons, radio buttons, toggle buttons, and the selected item in popup menus, when the specified text is clipped because it is too long for the **uicontrol**, an ellipsis (...) is appended to the text in the active GUI to indicate that it has been clipped.

For **editable text**, this property value is set to the string entered by the user.

Setting the String Property to a Reserved Word

To set the `String` property to one of the reserved words `default`, `remove`, or `factory`, you must precede the word with the backslash character. For example,

```
h = uicontrol('Style','edit','String','\default');
```

For more information see [Setting Default Values](#) in the MATLAB documentation.

style	{pushbutton} togglebutton
radiobutton	checkbox edit text slider
frame	listbox popupmenu

*Style of **uicontrol** object to create.* The `Style` property specifies the kind of **uicontrol** to create. See the **uicontrol** [Description](#) section for information on each type.

Tag	string (GUIDE sets this property)
------------	-----------------------------------

User-specified object label. The `Tag` property provides a means to identify graphics objects with a user-specified label. This is particularly useful when constructing interactive graphics programs that would otherwise need to define object handles as global variables or pass them as arguments between callback routines. You can define `Tag` as any string.

TooltipString	string
----------------------	--------

Content of tooltip for object. The `TooltipString` property specifies the text of the tooltip associated with the `uicontrol`. When the user moves the mouse pointer over the control and leaves it there, the tooltip is displayed.

Type string (read only)

Class of graphics object. For `uicontrol` objects, `Type` is always the string `'uicontrol'`.

UIContextMenu handle

Associate a context menu with `uicontrol`. Assign this property the handle of a `uicontextmenu` object. MATLAB displays the context menu whenever you right-click over the `uicontrol`. Use the `uicontextmenu` function to create the context menu.

Units {pixels} | normalized | inches | centimeters | points | characters (GUIDE default: normalized)

Units of measurement. MATLAB uses these units to interpret the `Extent` and `Position` properties. All units are measured from the lower-left corner of the parent object.

- Normalized units map the lower-left corner of the parent object to (0,0) and the upper-right corner to (1.0,1.0).
- pixels, inches, centimeters, and points are absolute units (1 point = 1/72 inch).
- Character units are characters using the default system font; the width of one character is the width of the letter x, the height of one character is the distance between the baselines of two lines of text.

If you change the value of `Units`, it is good practice to return it to its default value after completing your computation so as not to affect other functions that assume `Units` is set to the default value.

UserData matrix

User-specified data. Any data you want to associate with the `uicontrol` object. MATLAB does not use this data, but you can access it using `set` and `get`.

Value scalar or vector

Current value of `uicontrol`. The `uicontrol` style determines the possible values this property can have:

- Check boxes set `Value` to `Max` when they are on (when selected) and `Min` when off (not selected).
- List boxes set `Value` to a vector of indices corresponding to the selected list entries, where 1 corresponds to the first item in the list.
- Pop-up menus set `Value` to the index of the item selected, where 1 corresponds to the first item in the menu. The [Examples section](#) shows how to use the `Value` property to determine which item has been selected.

- Radio buttons set `Value` to `Max` when they are on (when selected) and `Min` when off (not selected).
- Sliders set `Value` to the number indicated by the slider bar.
- Toggle buttons set `Value` to `Max` when they are down (selected) and `Min` when up (not selected).
- Editable text, push buttons, and static text do not set this property.

Set the `Value` property either interactively with the mouse or through a call to the `set` function. The display reflects changes made to `Value`.

`visible` {on} | off

Uicontrol *visibility*. By default, all uicontrols are visible. When set to `off`, the **uicontrol** is not visible, but still exists and you can query and set its properties.

Note Setting `visible` to `off` for uicontrols that are not displayed initially in the GUI, can result in faster startup time for the GUI.

◀ **uicontrol** uigetdir ▶

© 1994–2005 The MathWorks, Inc. • [Terms of Use](#) • [Patents](#) • [Trademarks](#)