
Vanilla Documentation

Release 0.1

Tal Leming

Apr 19, 2019

Contents

1 Concepts	1
2 Objects	3
3 Indices and tables	15
Python Module Index	17

CHAPTER 1

Concepts

2.1 Windows

2.1.1 Window

Window.**assignToDocument** (*document*)

Add this window to the list of windows associated with a document.

document should be a *NSDocument* instance.

Window.**getNSWindow** ()

Return the *NSWindow* that this Vanilla object wraps.

Window.**getNSWindowController** ()

Return an *NSWindowController* for the *NSWindow* that this Vanilla object wraps, creating a one if needed.

Window.**open** ()

Open the window.

Window.**close** ()

Close the window.

Once a window has been closed it can not be re-opened.

Window.**hide** ()

Hide the window.

Window.**show** ()

Show the window if it is hidden.

Window.**makeKey** ()

Make the window the key window.

Window.**makeMain** ()

Make the window the main window.

Window.**setTitle** (*title*)

Set the title in the window's title bar.

title should be a string.

Window.**getTitle** ()

The title in the window's title bar.

Window.**select** ()

Select the window if it is not the currently selected window.

Window.**isVisible** ()

A boolean value representing if the window is visible or not.

Window.**getPosSize** ()

A tuple of form (*left, top, width, height*) representing the window's position and size.

Window.**setPosSize** (*posSize, animate=True*)

Set the position and size of the window.

posSize A tuple of form (*left, top, width, height*).

Window.**center** ()

Center the window within the screen.

Window.**move** (*x, y, animate=True*)

Move the window by **x** units and **y** units.

Window.**resize** (*width, height, animate=True*)

Change the size of the window to **width** and **height**.

Window.**setDefaultButton** (*button*)

Set the default button in the window.

button will be bound to the Return and Enter keys.

Window.**bind** (*event, callback*)

Bind a callback to an event.

event A string representing the desired event. The options are:

<i>"should close"</i>	Called when the user attempts to close the window. This must return a bool indicating if the window should be closed or not.
<i>"close"</i>	Called immediately before the window closes.
<i>"move"</i>	Called immediately after the window is moved.
<i>"resize"</i>	Called immediately after the window is resized.
<i>"became main"</i>	Called immediately after the window has become the main window.
<i>"resigned main"</i>	Called immediately after the window has lost its main window status.
<i>"became key"</i>	Called immediately after the window has become the key window.
<i>"resigned key"</i>	Called immediately after the window has lost its key window status.

For more information about main and key windows, refer to the Cocoa 'documentation <<http://developer.apple.com/documentation/Cocoa/Conceptual/WinPanel/Concepts/ChangingMainKeyWindow.html>> ' on the subject.

callback The callback that will be called when the event occurs. It should accept a *sender* argument which will be the Window that called the callback.:


```

from vanilla import Window

class WindowBindDemo(object):

    def __init__(self):
        self.w = Window((200, 200))
        self.w.bind("move", self.windowMoved)
        self.w.open()

    def windowMoved(self, sender):
        print("window moved!", sender)

WindowBindDemo()

```

Window.**unbind** (*event*, *callback*)

Unbind a callback from an event.

event A string representing the desired event. Refer to *bind* for the options.

callback The callback that has been bound to the event.

Window.**addToolbar** (*toolbarIdentifier*, *toolbarItems*, *addStandardItems=True*)

Add a toolbar to the window.

toolbarIdentifier A string representing a unique name for the toolbar.

toolbarItems An ordered list of dictionaries containing the following items:

<i>itemIdentifier</i>	A unique string identifier for the item. This is only used internally.
<i>label</i> (optional)	The text label for the item. Defaults to <i>None</i> .
<i>paletteLabel</i> (optional)	The text label shown in the customization palette. Defaults to <i>label</i> .
<i>toolTip</i> (optional)	The tool tip for the item. Defaults to <i>label</i> .
<i>imagePath</i> (optional)	A file path to an image. Defaults to <i>None</i> .
<i>imageNamed</i> (optional)	The name of an image already loaded as a <i>NSImage</i> by the application. Defaults to <i>None</i> .
<i>imageObject</i> (optional)	A <i>_NSImage_</i> object. Defaults to <i>None</i> .
<i>selectable</i> (optional)	A boolean representing if the item is selectable or not. The default value is <i>_False_</i> . For more information on selectable toolbar items, refer to Apple's documentation
<i>view</i> (optional)	A <i>NSView</i> object to be used instead of an image. Defaults to <i>None</i> .
<i>visibleByDefault</i> (optional)	If the item should be visible by default pass <i>True</i> to this argument. If the item should be added to the toolbar only through the customization palette, use a value of <i>_False_</i> . Defaults to <i>_True_</i> .

addStandardItems A boolean, specifying whether the standard Cocoa toolbar items should be added. Defaults to *True*. If you set it to *False*, you must specify any standard items manually in *toolbarItems*, by using the constants from the *AppKit* module:

<i>NSToolbarSeparatorItemIdentifier</i>	The Separator item.
<i>NSToolbarSpaceItemIdentifier</i>	The Space item.
<i>NSToolbarFlexibleSpaceItemIdentifier</i>	The Flexible Space item.
<i>NSToolbarShowColorsItemIdentifier</i>	The Colors item. Shows the color panel.
<i>NSToolbarShowFontsItemIdentifier</i>	The Fonts item. Shows the font panel.
<i>NSToolbarCustomizeToolbarItemIdentifier</i>	The Customize item. Shows the customization palette.
<i>NSToolbarPrintItemIdentifier</i>	The Print item. Refer to Apple's <i>NSToolbarItem</i> documentation for more information.

Returns a dictionary containing the created toolbar items, mapped by itemIdentifier.

2.1.2 FloatingWindow

`FloatingWindow.assignToDocument (document)`

Add this window to the list of windows associated with a document.

document should be a *NSDocument* instance.

`FloatingWindow.getNSWindow ()`

Return the *NSWindow* that this Vanilla object wraps.

`FloatingWindow.getNSWindowController ()`

Return an *NSWindowController* for the *NSWindow* that this Vanilla object wraps, creating a one if needed.

`FloatingWindow.open ()`

Open the window.

`FloatingWindow.close ()`

Close the window.

Once a window has been closed it can not be re-opened.

`FloatingWindow.hide ()`

Hide the window.

`FloatingWindow.show ()`

Show the window if it is hidden.

`FloatingWindow.makeKey ()`

Make the window the key window.

`FloatingWindow.makeMain ()`

Make the window the main window.

`FloatingWindow.setTitle (title)`

Set the title in the window's title bar.

title should be a string.

`FloatingWindow.getTitle ()`

The title in the window's title bar.

`FloatingWindow.select ()`

Select the window if it is not the currently selected window.

`FloatingWindow.isVisible()`

A boolean value representing if the window is visible or not.

`FloatingWindow.getPosSize()`

A tuple of form *(left, top, width, height)* representing the window's position and size.

`FloatingWindow.setPosSize(posSize, animate=True)`

Set the position and size of the window.

posSize A tuple of form *(left, top, width, height)*.

`FloatingWindow.center()`

Center the window within the screen.

`FloatingWindow.move(x, y, animate=True)`

Move the window by **x** units and **y** units.

`FloatingWindow.resize(width, height, animate=True)`

Change the size of the window to **width** and **height**.

`FloatingWindow.setDefaultButton(button)`

Set the default button in the window.

button will be bound to the Return and Enter keys.

`FloatingWindow.bind(event, callback)`

Bind a callback to an event.

event A string representing the desired event. The options are:

<i>"should close"</i>	Called when the user attempts to close the window. This must return a bool indicating if the window should be closed or not.
<i>"close"</i>	Called immediately before the window closes.
<i>"move"</i>	Called immediately after the window is moved.
<i>"resize"</i>	Called immediately after the window is resized.
<i>"became main"</i>	Called immediately after the window has become the main window.
<i>"resigned main"</i>	Called immediately after the window has lost its main window status.
<i>"became key"</i>	Called immediately after the window has become the key window.
<i>"resigned key"</i>	Called immediately after the window has lost its key window status.

For more information about main and key windows, refer to the Cocoa 'documentation' <<http://developer.apple.com/documentation/Cocoa/Conceptual/WinPanel/Concepts/ChangingMainKeyWindow.html>> ' on the subject.

callback The callback that will be called when the event occurs. It should accept a *sender* argument which will be the Window that called the callback.:

```
class WindowBindDemo(object):

    def __init__(self):
        self.w = Window((200, 200))
        self.w.bind("move", self.windowMoved)
        self.w.open()

    def windowMoved(sender):
```

(continues on next page)

(continued from previous page)

```

print("window moved!", sender)

WindowBindDemo()

```

`FloatingWindow.unbind(event, callback)`

Unbind a callback from an event.

event A string representing the desired event. Refer to *bind* for the options.

callback The callback that has been bound to the event.

`FloatingWindow.addToolbar(toolbarIdentifier, toolbarItems, addStandardItems=True)`

Add a toolbar to the window.

toolbarIdentifier A string representing a unique name for the toolbar.

toolbarItems An ordered list of dictionaries containing the following items:

<i>itemIdentifier</i>	A unique string identifier for the item. This is only used internally.
<i>label</i> (optional)	The text label for the item. Defaults to <i>None</i> .
<i>paletteLabel</i> (optional)	The text label shown in the customization palette. Defaults to <i>label</i> .
<i>toolTip</i> (optional)	The tool tip for the item. Defaults to <i>label</i> .
<i>imagePath</i> (optional)	A file path to an image. Defaults to <i>None</i> .
<i>imageNamed</i> (optional)	The name of an image already loaded as a <i>NSImage</i> by the application. Defaults to <i>None</i> .
<i>imageObject</i> (optional)	A <i>_NSImage_</i> object. Defaults to <i>None</i> .
<i>selectable</i> (optional)	A boolean representing if the item is selectable or not. The default value is <i>_False_</i> . For more information on selectable toolbar items, refer to Apple's documentation
<i>view</i> (optional)	A <i>NSView</i> object to be used instead of an image. Defaults to <i>None</i> .
<i>visibleByDefault</i> (optional)	If the item should be visible by default pass <i>True</i> to this argument. If the item should be added to the toolbar only through the customization palette, use a value of <i>_False_</i> . Defaults to <i>_True_</i> .

addStandardItems A boolean, specifying whether the standard Cocoa toolbar items should be added. Defaults to *True*. If you set it to *False*, you must specify any standard items manually in *toolbarItems*, by using the constants from the `AppKit` module:

<i>NSToolbarSeparatorItemIdentifier</i>	The Separator item.
<i>NSToolbarSpaceItemIdentifier</i>	The Space item.
<i>NSToolbarFlexibleSpaceItemIdentifier</i>	The Flexible Space item.
<i>NSToolbarShowColorsItemIdentifier</i>	The Colors item. Shows the color panel.
<i>NSToolbarShowFontsItemIdentifier</i>	The Fonts item. Shows the font panel.
<i>NSToolbarCustomizeToolbarItemIdentifier</i>	The Customize item. Shows the customization palette.
<i>NSToolbarPrintItemIdentifier</i>	The Print item. Refer to Apple's <i>NSToolbarItem</i> documentation for more information.

Returns a dictionary containing the created toolbar items, mapped by `itemIdentifier`.

2.1.3 Sheet

Sheet.**assignToDocument** (*document*)

Add this window to the list of windows associated with a document.

document should be a *NSDocument* instance.

Sheet.**getNSWindow** ()

Return the *NSWindow* that this Vanilla object wraps.

Sheet.**getNSWindowController** ()

Return an *NSWindowController* for the *NSWindow* that this Vanilla object wraps, creating a one if needed.

Sheet.**open** ()

Open the window.

Sheet.**close** ()

Close the window.

Once a window has been closed it can not be re-opened.

Sheet.**hide** ()

Hide the window.

Sheet.**show** ()

Show the window if it is hidden.

Sheet.**makeKey** ()

Make the window the key window.

Sheet.**makeMain** ()

Make the window the main window.

Sheet.**setTitle** (*title*)

Set the title in the window's title bar.

title should be a string.

Sheet.**getTitle** ()

The title in the window's title bar.

Sheet.**select** ()

Select the window if it is not the currently selected window.

Sheet.**isVisible** ()

A boolean value representing if the window is visible or not.

Sheet.**getPosSize** ()

A tuple of form (*left, top, width, height*) representing the window's position and size.

Sheet.**setPosSize** (*posSize, animate=True*)

Set the position and size of the window.

posSize A tuple of form (*left, top, width, height*).

Sheet.**center** ()

Center the window within the screen.

Sheet.**move** (*x, y, animate=True*)

Move the window by **x** units and **y** units.

Sheet.**resize** (*width, height, animate=True*)

Change the size of the window to **width** and **height**.

Sheet.**setDefaultButton** (*button*)

Set the default button in the window.

button will be bound to the Return and Enter keys.

Sheet.**bind** (*event, callback*)

Bind a callback to an event.

event A string representing the desired event. The options are:

"should close"	Called when the user attempts to close the window. This must return a bool indicating if the window should be closed or not.
"close"	Called immediately before the window closes.
"move"	Called immediately after the window is moved.
"resize"	Called immediately after the window is resized.
"became main"	Called immediately after the window has become the main window.
"resigned main"	Called immediately after the window has lost its main window status.
"became key"	Called immediately after the window has become the key window.
"resigned key"	Called immediately after the window has lost its key window status.

For more information about main and key windows, refer to the Cocoa 'documentation' <<http://developer.apple.com/documentation/Cocoa/Conceptual/WinPanel/Concepts/ChangingMainKeyWindow.html>> ' on the subject.

callback The callback that will be called when the event occurs. It should accept a *sender* argument which will be the Window that called the callback.:

```
class WindowBindDemo(object):

    def __init__(self):
        self.w = Window((200, 200))
        self.w.bind("move", self.windowMoved)
        self.w.open()

    def windowMoved(self, sender):
```

(continues on next page)

(continued from previous page)

```

    print("window moved!", sender)

WindowBindDemo()

```

Sheet.**unbind**(*event, callback*)

Unbind a callback from an event.

event A string representing the desired event. Refer to *bind* for the options.

callback The callback that has been bound to the event.

Sheet.**addToolbar**(*toolbarIdentifier, toolbarItems, addStandardItems=True*)

Add a toolbar to the window.

toolbarIdentifier A string representing a unique name for the toolbar.

toolbarItems An ordered list of dictionaries containing the following items:

<i>itemIdentifier</i>	A unique string identifier for the item. This is only used internally.
<i>label</i> (optional)	The text label for the item. Defaults to <i>None</i> .
<i>paletteLabel</i> (optional)	The text label shown in the customization palette. Defaults to <i>label</i> .
<i>toolTip</i> (optional)	The tool tip for the item. Defaults to <i>label</i> .
<i>imagePath</i> (optional)	A file path to an image. Defaults to <i>None</i> .
<i>imageName</i> (optional)	The name of an image already loaded as a <i>NSImage</i> by the application. Defaults to <i>None</i> .
<i>imageObject</i> (optional)	A <i>_NSImage_</i> object. Defaults to <i>None</i> .
<i>selectable</i> (optional)	A boolean representing if the item is selectable or not. The default value is <i>_False_</i> . For more information on selectable toolbar items, refer to Apple's documentation
<i>view</i> (optional)	A <i>NSView</i> object to be used instead of an image. Defaults to <i>None</i> .
<i>visibleByDefault</i> (optional)	If the item should be visible by default pass <i>True</i> to this argument. If the item should be added to the toolbar only through the customization palette, use a value of <i>_False_</i> . Defaults to <i>_True_</i> .

addStandardItems A boolean, specifying whether the standard Cocoa toolbar items should be added. Defaults to *True*. If you set it to *False*, you must specify any standard items manually in *toolbarItems*, by using the constants from the *AppKit* module:

<i>NSToolbarSeparatorItemIdentifier</i>	The Separator item.
<i>NSToolbarSpaceItemIdentifier</i>	The Space item.
<i>NSToolbarFlexibleSpaceItemIdentifier</i>	The Flexible Space item.
<i>NSToolbarShowColorsItemIdentifier</i>	The Colors item. Shows the color panel.
<i>NSToolbarShowFontsItemIdentifier</i>	The Fonts item. Shows the font panel.
<i>NSToolbarCustomizeToolbarItemIdentifier</i>	The Customize item. Shows the customization palette.
<i>NSToolbarPrintItemIdentifier</i>	The Print item. Refer to Apple's <i>NSToolbarItem</i> documentation for more information.

Returns a dictionary containing the created toolbar items, mapped by itemIdentifier.

2.1.4 Drawer

2.2 Layout Views

2.2.1 Group

2.2.2 ScrollView

2.2.3 SplitView

2.2.4 Box

2.2.5 HorizontalLine

2.2.6 VerticalLine

2.3 Data Views

2.3.1 List

2.3.2 List Item Cells

2.3.3 ImageView

2.3.4 LevelIndicator

2.4 Buttons

2.4.1 Button

2.4.2 SquareButton

2.4.3 ImageButton

2.4.4 GradientButton

2.4.5 HelpButton

2.4.6 SegmentedButton

2.5 Inputs

2.5.1 TextEditor

2.5.2 TextBox

2.5.3 EditText

2.5.4 SecureEditText

2.5.5 DatePicker

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

V

vanilla, 3

A

addToolBar() (*vanilla.FloatingWindow method*), 8
addToolBar() (*vanilla.Sheet method*), 11
addToolBar() (*vanilla.Window method*), 5
assignToDocument() (*vanilla.FloatingWindow method*), 6
assignToDocument() (*vanilla.Sheet method*), 9
assignToDocument() (*vanilla.Window method*), 3

B

bind() (*vanilla.FloatingWindow method*), 7
bind() (*vanilla.Sheet method*), 10
bind() (*vanilla.Window method*), 4

C

center() (*vanilla.FloatingWindow method*), 7
center() (*vanilla.Sheet method*), 10
center() (*vanilla.Window method*), 4
close() (*vanilla.FloatingWindow method*), 6
close() (*vanilla.Sheet method*), 9
close() (*vanilla.Window method*), 3

G

getNSWindow() (*vanilla.FloatingWindow method*), 6
getNSWindow() (*vanilla.Sheet method*), 9
getNSWindow() (*vanilla.Window method*), 3
getNSWindowController() (*vanilla.FloatingWindow method*), 6
getNSWindowController() (*vanilla.Sheet method*), 9
getNSWindowController() (*vanilla.Window method*), 3
getPosSize() (*vanilla.FloatingWindow method*), 7
getPosSize() (*vanilla.Sheet method*), 10
getPosSize() (*vanilla.Window method*), 4
getTitle() (*vanilla.FloatingWindow method*), 6
getTitle() (*vanilla.Sheet method*), 9
getTitle() (*vanilla.Window method*), 4

H

hide() (*vanilla.FloatingWindow method*), 6
hide() (*vanilla.Sheet method*), 9
hide() (*vanilla.Window method*), 3

I

isVisible() (*vanilla.FloatingWindow method*), 6
isVisible() (*vanilla.Sheet method*), 9
isVisible() (*vanilla.Window method*), 4

M

makeKey() (*vanilla.FloatingWindow method*), 6
makeKey() (*vanilla.Sheet method*), 9
makeKey() (*vanilla.Window method*), 3
makeMain() (*vanilla.FloatingWindow method*), 6
makeMain() (*vanilla.Sheet method*), 9
makeMain() (*vanilla.Window method*), 3
move() (*vanilla.FloatingWindow method*), 7
move() (*vanilla.Sheet method*), 10
move() (*vanilla.Window method*), 4

O

open() (*vanilla.FloatingWindow method*), 6
open() (*vanilla.Sheet method*), 9
open() (*vanilla.Window method*), 3

R

resize() (*vanilla.FloatingWindow method*), 7
resize() (*vanilla.Sheet method*), 10
resize() (*vanilla.Window method*), 4

S

select() (*vanilla.FloatingWindow method*), 6
select() (*vanilla.Sheet method*), 9
select() (*vanilla.Window method*), 4
setDefaultButton() (*vanilla.FloatingWindow method*), 7
setDefaultButton() (*vanilla.Sheet method*), 10
setDefaultButton() (*vanilla.Window method*), 4

`setSize()` (*vanilla.FloatingWindow method*), 7
`setSize()` (*vanilla.Sheet method*), 10
`setSize()` (*vanilla.Window method*), 4
`setTitle()` (*vanilla.FloatingWindow method*), 6
`setTitle()` (*vanilla.Sheet method*), 9
`setTitle()` (*vanilla.Window method*), 3
`show()` (*vanilla.FloatingWindow method*), 6
`show()` (*vanilla.Sheet method*), 9
`show()` (*vanilla.Window method*), 3

U

`unbind()` (*vanilla.FloatingWindow method*), 8
`unbind()` (*vanilla.Sheet method*), 11
`unbind()` (*vanilla.Window method*), 5

V

`vanilla` (*module*), 3, 6, 9, 14