

MiscBackWindow

Inherits From: Window : Responder : Object

Declared In: misckit/MiscBackWindow.h

Class Description

MiscBackWindow is a Window subclass that can be used as a full screen backdrop. It can be moved from one window tier to another at will and it has no visible frame. It is most useful for screen savers, applications backdrops and foreground slide shows. The ability to place a MiscBackWindow into the top most window tier means it can be placed in front of everything, including the dock, menus and alert panels. One should be a bit careful with this ability because if the application does not examine mouse or keyboard events as a signal for putting the MiscBackWindow into a more normal tier it will be difficult to regain control of the computer.

Note that MiscBackWindow has a Protocol so that MiscBackWindows may be controlled by remote applications.

The following code sample shows how a MiscBackWindow is created and used. In this case the event masks are set so that virtually any event will cause the method attention: to be sent to the object that contains this code. The object also will act as the MiscBackWindow delegate and thus receive reports of any change in the MiscBackWindow tier. It is then explicitly sent to the back of the backmost tier.

```
myWindow = [[MiscBackWindow allocFromZone: [self zone]] init];  
frameView = [myWindow contentView];  
[frameView setActionMask: NX_LMOUSEDOWNMASK | NX_RMOUSEDOWNMASK | \  
NX_MOUSEMOVEDMASK | NX_KEYDOWNMASK];  
  
[frameView setFirstMouse: YES];  
[frameView setTarget: self];  
[frameView setAction: @selector(attention:)];  
[frameView setFirstResponder];  
  
oldMask = [myWindow addToEventMask: NX_KEYDOWNMASK];  
[myWindow setDelegate: self];  
[myWindow makeBackdrop:self];  
[myWindow orderBack: self];
```

Instance Variables

int **currentTier**;

currentTier

Number of the window tier the MiscBackWindow currently resides in.

Method Types

Initialization - init

Tier Control

- setWindowTier: (int) tier
- makeFrontdrop:sender
- makeBackdrop:sender

Modified Window methods

- + getFrameRect: forContentRect:style:
- + getContentRect:forFrameRect:style:
- + minFrameWidth:forStyle:buttonMask:
- setDelegate:

Blocked Window methods

- initContent:

Archiving

- awake
- read:
- write:

Class Methods

getContentRect: forFrameRect:style:

+ **getContentRect:**(NXRect *)*cRect* **forFrameRect:**(const NXRect *)*fRect* **style:**(int)*aStyle*

Overrides a Window method to set the content rectangle to the area of the current screen. Method is not called directly.

Returns **self**.

See also: + **minFrameWidth:forStyle:buttonMask:**, + **getFrameRect:forContentRect:style:**

minFrameWidth:forStyle:buttonMask:

+ (NXCoord)**minFrameWidth:**(const char *)*aTitle* **forStyle:**(int)*aStyle* **buttonMask:**(int)*aMask*

Overrides a Window method. Method is not called directly. Returns the width of the current screen.

See also: + **getFrameRect:forContentRect:style:**, + **getContentRect: forFrameRect:style:**

getFrameRect:forContentRect:style:

+ **getFrameRect:**(NXRect *)*fRect* **forContentRect:**(const NXRect *)*cRect* **style:**(int)*aStyle*

Overrides a Window method to set the frame rectangle to the area of the current screen. Method is not called directly.

Returns **self**.

See also: + **minFrameWidth:forStyle:buttonMask:**, + **getContentRect: forFrameRect:style:**

Instance Methods

awake:

- **awake**

Executed when an object completes unarchiving. Returns **self**.

See also: - **read:**, - **write:**

init

- **init**

Initialize a MiscBackWindow object. The window frame is NX_TOKENSTYLE, buffering is NX_BUFFERED and there are no buttons. The *contentView* is a FramingView with an MISC_FV_PLAIN border (See MiscFramingView), the default background gray of both window and view is NX_BLACK, and the window is initially placed in the MISC_BACKGROUNDTIER . Note the importance of the MiscFramingView: it not only allows for easy resizing and justification of images in the background, but it also supports mouse and keyboard events in a target-action protocol which makes it easy to get control of windows that are placed in a high tier. Returns **self**.

initWithContent:

- **initWithContent:**(const NXRect *)*contentRect*

This Window method is not to be used. Generates an error message if used.

makeBackdrop:

- **makeBackdrop:***sender*

Puts the window into the rearmost window tier. At the same time it removes NX_MOUSEMOVEDMASK to the event mask. Returns **self**.

See also: - **makeFrontdrop:**, - **setWindowTier:**

makeFrontdrop:

- **makeFrontdrop:***sender*

Puts the window into the frontmost window tier. At the same time it adds NX_MOUSEMOVEDMASK to the event mask. It is up to the user of this to retain some means of getting it *out* of the frontmost tier. Caveat Emptor. Returns **self**.

See also: - **setWindowTier:**, - **makeBackdrop:**

read:

- **read:**(NXTypedStream *)*stream*

Reads the object from the typed stream *stream*. Returns **self**.

See also: - `write:`, - `awake`

setDelegate:

- `setDelegate:`*anObject*

Sets the window delegate. For some reason the `Window` method of the same name does not function in this class under 3.0. Returns **self**.

setWindowTier:

- `setWindowTier:`*(int)tier*

Puts the window into the specified window *tier*. If it is used to place the window into a tier above menu's or panels, it is up to the user to retain some means of getting it *out* of that tier. *Caveat Emptor*. The following are defined for convenience:

MISC_SAVERTIER	Highest window tier
MISC_BACKGROUND TIER	Lowest window tier

Returns **self**.

See also: - `makeFrontdrop:`, - `makeBackdrop:`

write:

- `write:`*(NXTypedStream *)stream*

Writes the object to the typed stream *stream*. Returns **self**.

See also: - `read:`, - `awake`

Delegate Methods

windowDidChangeTierTo:

- `windowDidChangeTierTo:(int)tier`

Sent to the delegate when a window tier change occurs. Returns **self**.