NS_DEV_DOCFOR:objc_class:SavePanel;,MiscAppIconAnimator

Inherits From: NSObject

Declared In:

Class Description

MiscAppIconAnimator has been modeled after Fone.app (v1.0) by Jim Million, October 1992.
The MiscAppIconAnimator is a class defining an object that provides a convenient way to animate the icon for an application. Animation requires an array of images/icons to display in the icon, an array indicating the pattern/order to display the images, and the number of times to loop through the pattern . Other information that may be specified include: the image to display when the animation sequence finishes, the image to display when animation is stopped, and the time interval between image changes.

The display of images into the Application Icon will not work until the Application is fully initialized, and the icon is in place.

The animation sequence uses TimerEvents to loop through the animation pattern array. The pattern array contains the order in which to display the images. Each entry in the pattern array is an index into the array of images.

The **startAnimation:** method starts the animation sequence in the icon. The **stopAnimation** method is used to stop the animation sequence. The **initWithImageArray: animationPattern: andPatternLoops:** method is the designated initializer for the MiscAppIconAnimator object.

Instance Variables

NXImage ***appTile**;
View ***appIconContentView**;

NSMutableArray ***animationPattern**;
NSMutableArray ***imageArray**;
int **patternLoops**;
float **timeInterval**;
int **indexOfEndIcon**;
int **indexOfStopIcon**;
DPSTimedEntry **timedEntry**;
int **patternIterator**;
int **loopCount**;
int **patternEnd**;
int **offset**;

appTile	The application's tile "NXAppTile".
appIconContentView	The contentView of the application's icon.
animationPattern	Array of indices into the imageArray, indicating the order to display the images.
imageArray	Array of images to be placed in the icon.
patternLoops	The number of times that the animationPattern will be looped through.
timeInterval	The time interval between image changes. (default = 0.2 seconds)
indexOfEndIcon	Index into the imageArray, indicating what image to display when the animation sequence ends. (default = 1)
indexOfStopIcon	Index into the imageArray, indicating what image to display when the animation sequence is stopped. (default = 0)
timedEntry	The identifier of the timed entry.
patternIterator	The current pattern entry being animated.
loopCount	The current number of loops through the pattern.
patternEnd	The number of entries in the animation pattern.
offset	Used to limit the display of the last image in the pattern to the first loop through.

Method Types

Initializing an MiscAppIconAnimator
- init
- initWithImageArray:animationPattern:andPatternLoops:

Configuring an MiscAppIconAnimator
- imageArray
- setimageArray:
- animationPattern
- setAnimationPattern:
- patternLoops
- setPatternLoops:
- timeInterval
- setTimeInterval:
- indexOfEndIcon
- setIndexOfEndIcon:
- indexOfStopIcon
- setIndexOfStopIcon:

Controlling the animation
- startAnimation:
- stopAnimation

Displaying an Image in the icon
- displayImage:
- displayImageAtIndex:

Archiving
- encodeWithCoder:
- initWithCoder:

Distributed Object methods
- encodeRemotelyFor:freeAfterEncoding:isBycopy:
- encodeUsing:
- decodeUsing:

Internal methods
- animateIcon
- cornerPointForImage:
- removeTimedEntry

Instance Methods

animateIcon

- (void)**animateIcon**

Control the animation of images into the application icon. This method is called at the end of each timer interval. The next image from **imageArray** identified in the **animationPattern** is displayed, and then processing continues based on the input parameters. The timer may continue, or timing and the animation sequence may end. When the animation sequence ends, a designated image is displayed based on **indexOfEndIcon**.

See also: - **animationPattern,** - **displayImage:,** - **imageArray,** - **indexOfEndIcon,** - **patternLoops,** - **removeTimedEntry,** - **startAnimation:,** - **stopAnimation**

NS_DEV_DOCFOR:objc_method:[SavePanel-doesTreatFilePackagesAsDirectories];,animationPattern

- (NSArray *)**animationPattern**

Returns the NSMutableArray set by **setAnimationPattern:**. This array contains the pattern used in determining what order to display images in the animation sequence.

See also: - **setAnimationPattern:**

cornerPointForImage:

- (NXPoint)**cornerPointForImage:**(NXImage*)*anImage*

Return the point that should be used as the corner for *anImage*. This is used to make sure the image is centered on the application icon.

See also: - **displayImage:**

decodeUsing:

- **decodeUsing:**(id <NXEncoding>)*portal*

Decode MiscAppIconAnimator instance variables from *portal* after passing through Distributed Objects.

See also: - **encodeUsing:**, - **encodeRemotelyFor:freeAfterEncoding:isBycopy:**

displayImage:

- (void)**displayImage:**(NXImage*)*anImage*

Display *anImage* into the application icon.

See also: - **cornerPointForImage:**, - **displayImageAtIndex:**

displayImageAtIndex:

- (void)**displayImageAtIndex:**(int)*imageNum*

Display the image from the imageArray at index *imageNum* into the application icon.

See also: - **displayImage:**

encodeRemotelyFor:freeAfterEncoding:isBycopy:

- **encodeRemotelyFor:**(NXConnection *)*connection* **freeAfterEncoding:**(BOOL *)*flagp* **isBycopy:**(BOOL)*isBycopy*

Encode MiscAppIconAnimator onto *portal* for passing through Distributed Objects. For this to work with an NSObject, this method must also be defined for NSObject. This is done through a category placed in the MiscAppIconAnimator header file.

See also: - **decodeUsing:**, - **encodeUsing:**

encodeUsing:

- **encodeUsing:**(id <NXEncoding>)*portal*

Encode MiscAppIconAnimator instance variables onto *portal* for passing through Distributed Objects.

See also: - **decodeUsing:**, - **encodeRemotelyFor:freeAfterEncoding:isBycopy:**

encodeWithCoder:

- (void)**encodeWithCoder:**(NSCoder *)*aCoder*

Encode MiscAppIconAnimator instance variables using *aCoder* for archiving.

See also: - **initWithCoder:**

imageArray

- (NSArray*)**imageArray**

Returns the NSMutableArray set by **setImageArray:**. This array contains the images displayed in the animation sequence.

See also: - **setImageArray:**

indexOfEndIcon

- (int)**indexOfEndIcon**

Returns the indexOfEndIcon. The image associated with this index is displayed in the icon when the animation sequence ends.

See also: - **setIndexOfEndIcon:**

indexOfStopIcon

- (int)**indexOfStopIcon**

Returns the indexOfStopIcon. The image associated with this index is displayed in the icon when the animation sequence is stopped.

See also: - **setIndexOfStopIcon:**

NS_DEV_DOCFOR:objc_method:[SavePanel-alloc];,init

- **init**

This is an override of the superclass's designated initializer. Calls **initWithImageArray:animationPattern:andPatternLoops:** with empty arrays, and pattern loops of zero.

See also: - **initWithImageArray:animationPattern:andPatternLoops:**

initWithCoder:

- **initWithCoder:**(NSCoder *)*aDecoder*

Decode MiscAppIconAnimator instance variables using *aDecoder* for unarchiving.

See also: - **encodeWithCoder:**

NS_DEV_DOCFOR:objc_method:[SavePanel-alloc];,initWithImageArray:animationPattern:andPatternLoops:

- **initWithImageArray:**(NSArray *)*anArray* **animationPattern:**(NSArray *)*patternArray* **andPatternLoops:**(int)*loops*

This is the Designated initializer for the MiscAppIconAnimator class. Initializes the appTile and the appIconContentView. Sets the **imageArray**, **animationPattern**, and **patternLoops** from the provided input parameters.

See also: - **animationPattern**, - **imageArray**, - **patternLoops**, - **setAnimationPattern:**, - **setImageArray:**, - **setPatternLoops:**

patternLoops

- (int)**patternLoops**

Returns the number of loops through the **animationPattern** that should be executed during the animation sequence.

See also: - **setPatternLoops**

removeTimedEntry

- (void)**removeTimedEntry**

Removes the DPS timed entry if one is currently running.

See also:

setAnimationPattern:

- (void)**setAnimationPattern:**(NSArray*)*anArray*

Sets the current **animationPattern** array to *anArray*. Sets patternEnd to the number of entries in the array.

See also: - **animationPattern**

setImageArray:

- (void)**setImageArray:**(NSArray*)*anArray*

Sets the current **imageArray** array to *anArray*.

See also: - **imageArray**

setIndexOfEndIcon:

- (void)**setIndexOfEndIcon:**(int)*endIcon*

Sets the **indexOfEndIcon** to *endIcon*.

See also: - **indexOfEndIcon**

setIndexOfStopIcon:

- (void)**setIndexOfStopIcon:**(int)*stopIcon*

Sets the **indexOfStopIcon** to *stopIcon*.

See also: - **indexOfStopIcon**

setPatternLoops:

- (void)**setPatternLoops:**(int)*numLoops*

Sets the **patternLoops** to *numLoops*.

See also: - **patternLoops**

setTimeInterval:

- (void)**setTimeInterval:**(float)*interval*

Sets the **timeInterval** between image animations to *interval* seconds.

See also: - **timeInterval**

startAnimation:

- (void)**startAnimation:***sender*

Starts the animation sequence.

See also: - **stopAnimation**

stopAnimation

- (void)**stopAnimation**

Stops and resets the animation sequence. A designated image is displayed based on **indexOfStopIcon**.

See also: - **startAnimation:**

timeInterval

- (float)**timeInterval**

Returns the **timeInterval** between image animations in seconds.

See also: - **setTimeInterval:**