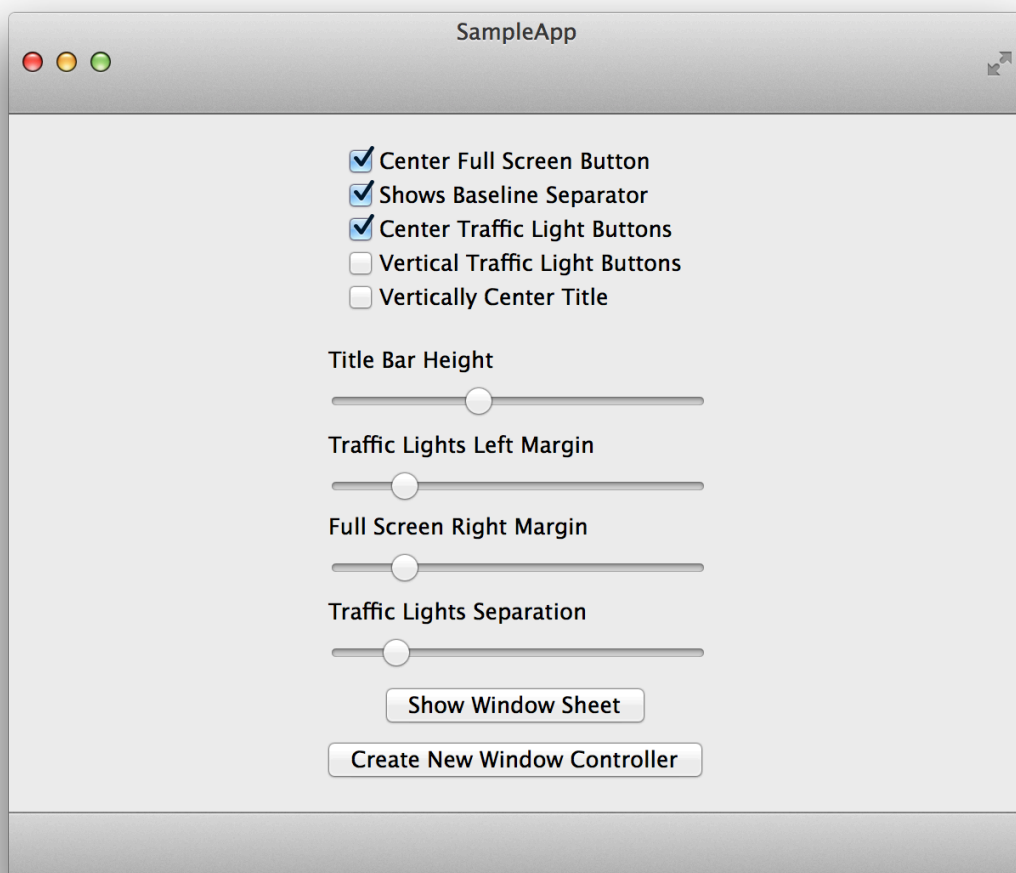

INAppStoreWindow

Title bar and traffic light customization for NSWindow `INAppStoreWindow` is an `NSWindow` subclass that was originally developed to mimic the appearance of the main window in the Mac App Store application introduced in OS X 10.6.6.

The MAS application has since transitioned away from this design, but `INAppStoreWindow` is still being actively developed to provide extensive additional customization options for `NSWindow` title bars.



Features of INAppStoreWindow:

- No private API usage and Mac App Store friendly.
- Fully customizable title bar view – add subviews (toolbars, buttons, etc.) and block based drawing for custom backgrounds

-
- Adjustable title bar height
 - Customization of the traffic light/fullscreen buttons for all button states.
 - Customization of the window title text.
 - Works on OS X versions 10.6-10.9
 - Compatible with full screen mode in OS X 10.7+

All of these great apps are using `INAppStoreWindow` in production!

Usage

ARC

`INAppStoreWindow` now requires ARC to compile. If your project does not use ARC, compile `INAppStoreWindow.m` with the `-fobjc-arc` linker flag.

Basic Configuration

Using `INAppStoreWindow` is as easy as changing the class of the `NSWindow` in Interface Builder, or simply by creating an instance of `INAppStoreWindow` in code (if you're doing it programmatically). I've included a sample project demonstrating how to use `INAppStoreWindow`.

NOTE: The title bar height is set to the standard window title height by default. You must set the 'titleBarHeight' property in order to increase the height of the title bar.

Some people seem to be having an issue where the title bar height property is not set properly when calling the method on an `NSWindow` without typecasting it to the `INAppStoreWindow` class. If you are experiencing this issue, do something like this (using a window controller, for example):

```
1 INAppStoreWindow *aWindow = (INAppStoreWindow*)[viewController window];
2 aWindow.titleBarHeight = 60.0;
```

Adding buttons and other controls to the title bar

Adding controls and other views to the title bar is simple. This can be done either programmatically or through Interface Builder. Here are examples of both methods:

Programmatically

```
1 // This code places a 100x100 button in the center of the title bar
  view
2 NSView *titleLabelView = self.window.titleLabelView;
3 NSSize buttonSize = CGSizeMake(100.f, 100.f);
4 CGRect buttonFrame = CGRectMake(NSMidX(titleBarView.bounds) - (
    buttonSize.width / 2.f), NSMidY(titleBarView.bounds) - (buttonSize.
    height / 2.f), buttonSize.width, buttonSize.height);
5 NSButton *button = [[NSButton alloc] initWithFrame:buttonFrame];
6 [button setTitle:@"A Button"];
7 [titleLabelView addSubview:button];
```

Interface Builder

NOTE: Even though the content layout for the title bar can be done in Interface Builder, you still need to use the below code to display the view created in IB in the title bar.

```
1 // self.titleLabelView is a an IBOutlet to an NSView that has been
  configured in IB with everything you want in the title bar
2 self.titleLabelView.frame = self.window.titleLabelView.bounds;
3 self.titleLabelView.autoresizingMask = NSViewWidthSizable |
  NSViewHeightSizable;
4 [self.window.titleLabelView addSubview:self.titleLabelView];
```

Centering the traffic light and full screen buttons

The vertical centering of the traffic light and full screen buttons can be controlled through two properties: `centerTrafficLightButtons` and `centerFullScreenButton`.

The traffic light buttons are vertically centered by default.

Hiding the title bar in fullscreen

You can tell `INAppStoreWindow` to hide when entering fullscreen mode, and reappear on exit. Just set the property `hideTitleBarInFullscreen` in order to hide it.

Padding the traffic lights and fullscreen buttons

The left padding of the traffic lights can be adjusted with `trafficLightButtonsLeftMargin` and the right padding of the fullscreen button can be adjusted with `fullscreenButtonRightMargin`.

Hiding the baseline (divider line between the titlebar and the content view)

The baseline divider can be hidden by setting `showsBaselineSeparator` to `NO`, the default value is `YES`.

Customizing traffic lights buttons

In order to customize these buttons, you would use `INWindowButton` class. You must create a separate instance for each button and provide your graphics for each state of the button. Currently the following states are supported:

- Active
- Active in not main window
- Inactive (disabled)
- Rollover
- Pressed

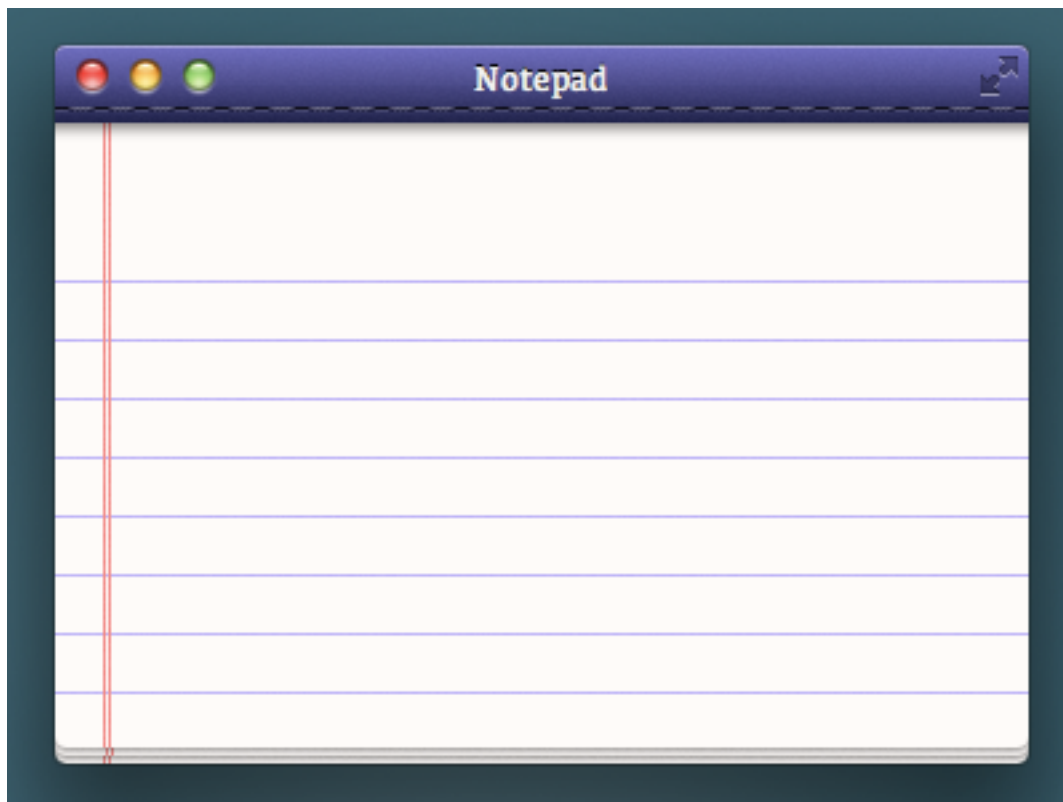
Please refer to `INWindowButton.h` header documentation for more details.

Customizing window's title appearance

You can enable title drawing by setting `showsTitle` property to `YES`. For `NSDocument` based apps, you can enable drawing the document proxy icon by setting `showsDocumentProxyIcon` property to `YES`. You can adjust appearance using `titleTextColor`, `inactiveTitleTextColor`, `titleTextShadow`, and `inactiveTitleTextShadow` properties. Also, you can enable title drawing in fullscreen by setting `showsTitleInFullscreen` property to `YES`.

Using your own drawing code

A lot of time and effort has gone into making the custom titlebar in `INAppStoreWindow` function just right, it would be a shame to have to re-implement all this work just to draw your own custom title bar. So `INAppStoreWindow` has a `titleBarDrawingBlock` property that can be set to a block containing your own drawing code!



```
1 [self.window setTitleBarDrawingBlock:^(BOOL drawsAsMainWindow, CGRect
2   drawingRect, CGRectEdge edge, CGPathRef clippingPath){
3   // Custom drawing code!
4 }];
```

This block gets passed some useful parameters like if the window is the main one(`drawsAsMainWindow`), the drawing rect of the title bar(`drawingRect`), and a pre-made clipping path with rounded corners at the top(`clippingPath`).

Setting the title bar colors

If you just want to adjust the color of the title bar without drawing the whole thing yourself, there are a few properties to help you do so:

```
1 self.window.titleBarStartColor = [NSColor colorWithCalibratedWhite:
2   0.6 alpha: 1.0];
3 self.window.titleBarEndColor   = [NSColor colorWithCalibratedWhite:
4   0.9 alpha: 1.0];
5 self.window.baselineSeparatorColor = [NSColor colorWithCalibratedWhite:
6   0.2 alpha: 1.0];
```

```
5 self.window.inactiveTitleBarEndColor = [NSColor
    colorWithCalibratedWhite: 0.95 alpha: 1.0];
6 self.window.inactiveTitleBarStartColor = [NSColor
    colorWithCalibratedWhite: 0.8 alpha: 1.0];
7 self.window.inactiveBaselineSeparatorColor = [NSColor
    colorWithCalibratedWhite: 0.4 alpha: 1.0];
```

Extensions

Additional extensions for `INAppStoreWindow` are provided in the **Extensions** subfolder.

NSDocument+INAppStoreWindowFixes

Add this category to your project to fix title bar layout for document based apps in response to `-[NSDocument updateChangeCount:]`. This fix was separated from the main `INAppStoreWindow` codebase because it swizzles methods on `NSDocument`.

INTitlebarView+CoreUIRendering

Add this category to your project to use CoreUI to render the system default title bar gradient, noise texture, and baseline separator. This will provide the most accurate visual results. You should **only** add this category to your project if you intend to release outside the App Store!

If you use this category you **must** define the preprocessor symbol `INAPPSTOREWINDOW_NO_COREUI` with a value of 1 in the build configuration for your Mac App Store version in order to exclude this code because it **uses private APIs and will result in rejection if included in a Mac App Store submission**.

When CoreUI rendering is not available, `INAppStoreWindow` will emulate the rendering of the system title bar as closely as possible.

C# Port (Unofficial)

`AppStoreWindow` is a pure MonoMac implementation of `INAppStoreWindow` by @ashokgelal.

Authors

`INAppStoreWindow` is maintained by Indragie Karunaratne, who could not have done it without **many awesome contributors**.

Licensing

INAppStoreWindow is licensed under the BSD license.