Kostenloses eBook

LERNEN

OSX

Free unaffiliated eBook created from
Stack Overflow contributors.

#osx

Inhaltsverzeichnis

Über ... 1

Kapitel 1: Erste Schritte mit osx ... 2

Bemerkungen ... 2

Examples ... 2

Überblick über Frameworks ... 2

Kapitel 2: Dateizuordnung ... 3

Examples ... 3

Legen Sie meine App als Standard-App für einen Dateityp fest ... 3

Erstellen Sie eine Verknüpfung mit neuen / benutzerdefinierten Dateitypen über Info.plist ... 3

Kapitel 3: Fordern Sie den Benutzer zur Eingabe einer Datei auf ... 5

Einführung ... 5

Examples ... 5

Dateien öffnen ... 5

Öffnen Sie eine beliebige Datei ... 5

Erlaubt das Öffnen mehrerer Dateien ... 5

Beschränkung auf bestimmte Dateitypen ... 5

Kapitel 4: NSFont ... 7

Einführung ... 7

Examples ... 7

NSFont-Objekt erstellen ... 7

Ziel c ... 7

Kapitel 5: NSMenuItem ... 8

Bemerkungen ... 8

Examples ... 8

Menüpunkte aktivieren ... 8

Manuelles Aktivieren der Menüelemente ... 8

Menüpunkte automatisch aktivieren ... 8

Unterstützende Standardmenüaktionen ... 9

Hinzufügen und Entfernen von Elementen zu einem Menü ... 9

Kapitel 6: NSRunLoop 10
Examples 10
einfache Daemon-Anwendung 10
Kapitel 7: NSStoryBoard 11
Examples 11
Öffnen Sie einen neuen Fenster-Controller 11
Kapitel 8: NSTextView 12
Einführung 12
Examples 12
NSTextView erstellen 12
Grafisch 12
Programmatisch 15
Ziel c 16
Kapitel 9: Umgebungsvariablen setzen 18
Einführung 18
Examples 18
Pfad hinzufügen 18
Credits 19

Über

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: osx

It is an unofficial and free osx ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official osx.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit osx

Bemerkungen

Dieses Tag dient zur Dokumentation der von Apple erstellten Mac-spezifischen Programmierschnittstellen, z. B. AppKit.

Examples

Überblick über Frameworks

Für macOS geschriebene Apps werden normalerweise mit Apples Frameworks geschrieben. Die Frameworks, die fast jede App verwenden wird, sind:

- AppKit - zum Erstellen und Verwalten von UI-Elementen
- Foundation - für allgemeine Objekte und Operationen außerhalb der Benutzeroberfläche

Es gibt andere gängige Frameworks, die in vielen, aber nicht in allen Apps verwendet werden:

- CoreData - zur Datenspeicherung
- Dispatch - zur Verwaltung mehrerer Threads
- CoreGraphics - zum Zeichnen von grafischen Aufgaben
- CoreAnimation - zur Animation von UI-Elementen

Erste Schritte mit osx online lesen: https://riptutorial.com/de/osx/topic/2818/erste-schritte-mit-osx

Kapitel 2: Dateizuordnung

Examples

Legen Sie meine App als Standard-App für einen Dateityp fest

```
- (NSString *) UTIforFileExtension:(NSString *) extension {
    NSString * UTIString = (NSString
*)UTTypeCreatePreferredIdentifierForTag(kUTTagClassFilenameExtension,
                                                                    (CFStringRef)extension,

                                                                    NULL);

    return [UTIString autorelease];
}

- (BOOL) setMyselfAsDefaultApplicationForFileExtension:(NSString *) fileExtension {
    OSStatus returnStatus = LSSetDefaultRoleHandlerForContentType (
                                                                   (CFStringRef) [self
UTIforFileExtension:fileExtension],
                                                                   kLSRolesAll,
                                                                   (CFStringRef) [[NSBundle
mainBundle] bundleIdentifier]
                                                                   );

    if (returnStatus != 0) {
        NSLog(@"Got an error when setting default application - %d", returnStatus);
        // Please see the documentation or LSInfo.h
        return NO;
    }

    return YES;
}
```

Quelle

Erstellen Sie eine Verknüpfung mit neuen / benutzerdefinierten Dateitypen über Info.plist

```
<key>CFBundleDocumentTypes</key>
<array>
    <dict>
        <key>CFBundleTypeIconFile</key>
        <string>Icon file for associated file</string>
        <key>CFBundleTypeName</key>
        <string>My file format</string>
        <key>CFBundleTypeRole</key>
        <string>Viewer</string> <!-- The value can be Editor, Viewer, Shell, or None. This key
is required.  -->
        <key>LSItemContentTypes</key>
        <array>
            <string>UTI of the file</string> <!-- Existing UTI or create a UTI for your new
file type -->
        </array>
```

```
        <key>LSHandlerRank</key>
        <string>Owner</string>
    </dict>
</array>
```

Quelle

Dateizuordnung online lesen: https://riptutorial.com/de/osx/topic/10926/dateizuordnung

Kapitel 3: Fordern Sie den Benutzer zur Eingabe einer Datei auf

Einführung

NSOpenPanel bietet eine API, die den Benutzer auffordert, eine Datei zu öffnen. Dieses Menü ist die Standardbenutzeroberfläche, die vom Menüelement Öffnen (O) angezeigt wird.

Examples

Dateien öffnen

Öffnen Sie eine beliebige Datei

```
NSOpenPanel *openPanel = [NSOpenPanel openPanel];
[openPanel beginWithCompletionHandler:^(NSInteger result) {
    NSURL *url = openPanel.URL;
    if (result == NSFileHandlingPanelCancelButton || !url) {
        return;
    }
    // do something with a URL
}];
```

Erlaubt das Öffnen mehrerer Dateien

```
NSOpenPanel *openPanel = [NSOpenPanel openPanel];
openPanel.allowsMultipleSelection = YES;
[openPanel beginWithCompletionHandler:^(NSInteger result) {
    NSArray <NSURL *>*urls = openPanel.URLs;
    // do things
}];
```

Beschränkung auf bestimmte Dateitypen

```
NSOpenPanel *openPanel = [NSOpenPanel openPanel];
openPanel.allowedFileTypes = @[@".png", @".jpg"];
[openPanel beginWithCompletionHandler:^(NSInteger result) {
    NSURL *url = openPanel.URL;
    if (result == NSFileHandlingPanelCancelButton || !url) {
        return;
    }
    // do something with a picture
}];
```

Fordern Sie den Benutzer zur Eingabe einer Datei auf online lesen:
https://riptutorial.com/de/osx/topic/9438/fordern-sie-den-benutzer-zur-eingabe-einer-datei-auf

Kapitel 4: NSFont

Einführung

NSFont ist das Objekt, das Mac-Anwendungen mit Glypheninformationen und Schriftcharakteristiken versorgt, die hauptsächlich für die Anzeige verwendet werden. Sie lernen, wie man NSFont-Objekte auf verschiedene Arten erstellt und verwendet, sowohl häufig als auch ungewöhnlich.

Examples

NSFont-Objekt erstellen

Die bevorzugte und häufigste Methode, ein NSFont-Objekt zu erstellen, ist folgende:

Ziel c

```
// Name is PostScript name of font; size is in points.
NSFont *essayFont = [NSFont fontWithName:@"Times New Roman" size:12.0];
```

NSFont online lesen: https://riptutorial.com/de/osx/topic/8881/nsfont

Kapitel 5: NSMenuItem

Bemerkungen

Die Dokumentation von Apple finden Sie hier:
https://developer.apple.com/library/mac/documentation/Cocoa/Reference/ApplicationKit/Classes/NSMenu

Examples

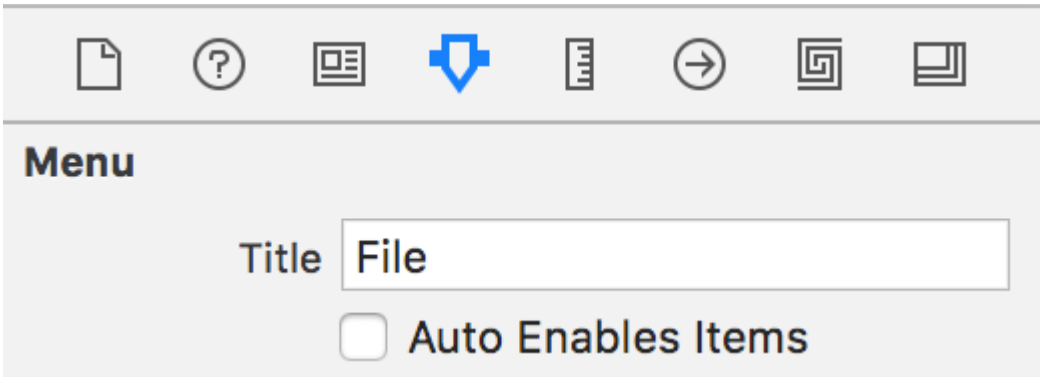
Menüpunkte aktivieren

Manuelles Aktivieren der Menüelemente

Um den aktivierten Status eines Menüelements manuell zu steuern, muss das Menü, in dem es enthalten ist, die automatische Aktivierung seiner Elemente deaktivieren

Menüs können die automatische Aktivierung auf zwei Arten deaktivieren:

1. Im Interface Builder



2. In Code

```
menu.autoenablesItems = false
```

Beide Mechanismen setzen die autoenablesItems -Eigenschaft in NSMenu .

Wenn das Menü nicht mehr aktiviert und deaktiviert wird, können die Menüelemente auf zwei verschiedene Arten programmiert werden

1. Im Interface Builder
2. In Code

```
menuItem.enabled = true
```

Menüpunkte automatisch aktivieren

Menüelemente können automatisch aktiviert werden, indem Menüelementaktionen mit dem Ersthelfer verbunden und die ausgelieferte Aktion für ein Objekt in der Responder-Kette implementiert wird, wie in den Dokumenten von Apple beschrieben.

Unterstützende Standardmenüaktionen

Menüs verhalten sich wie alle Standardsteuerelemente. Sie haben eine Aktion, die die aufzurufende Funktion ist, und ein Ziel , an das die Funktion gesendet werden soll. Wenn das Ziel auf ein Objekt festgelegt ist, wird die Aktionsmethode an das Zielobjekt gesendet, wenn ein Benutzer ein Menüelement auswählt. Wenn der Menüpunkt eine Aktion hat, aber kein Ziel, dann wird das Ziel dynamisch aus dem ersten Objekt aus dem Folgenden ausgewählt, das auf die Aktion reagiert:

1. Der Ersthelfer
2. Die Ansichtshierarchie
3. Fenster
4. Fenstersteuerung
5. `NSApplication`
6. `NSApplication.delegate`
7. `NSApplication.nextResponder`

Die Implementierung des standardmäßigen Menüelements Open (O) kann durch Implementieren der `openDocument` Methode für jedes Objekt in der obigen Liste erreicht werden.

```
- (IBAction)openDocument:(id)sender {

}
```

Hinzufügen und Entfernen von Elementen zu einem Menü

```
// add an item to a menu
menu.addItem(item)

// remove and item from a menu
menu.removeItem(item)
```

NSMenuItem online lesen: https://riptutorial.com/de/osx/topic/6038/nsmenuitem

Kapitel 6: NSRunLoop

Examples

einfache Daemon-Anwendung

Ein Daemon-Prozess führt ein Programm im Hintergrund aus, in der Regel ohne Benutzerinteraktion. Das folgende Beispiel zeigt, wie Sie einen Dämon erstellen und einen Listener registrieren, der alle geöffneten Anwendungen überwacht. Der Hauptteil ist der Funktionsaufruf `NSRunLoop.mainRunLoop().run()` , der den Dämon startet.

```
class MyObserver: NSObject
{
    override init() {
        super.init()

        // app listeners
        NSWorkspace.sharedWorkspace().notificationCenter.addObserver(self, selector:
"SwitchedApp:", name: NSWorkspaceDidActivateApplicationNotification, object: nil)
    }

    func SwitchedApp(notification: NSNotification!)
    {
        print(notification)
    }
}


let observer = MyObserver()

// simply to keep the command line tool alive - as a daemon process
NSRunLoop.mainRunLoop().run()
```

Sie können diesen Code auch als Grundlage für einen Serverprozess verwenden.

NSRunLoop online lesen: https://riptutorial.com/de/osx/topic/6667/nsrunloop

Kapitel 7: NSStoryBoard

Examples

Öffnen Sie einen neuen Fenster-Controller

Um ein neues Fenster zu öffnen, fügen Sie den folgenden Code an einer Stelle ein, an der Sie einen Verweis auf das neue Fenster (IE, den App-Delegaten) beibehalten können.

Schnell

```
let storyboard:NSStoryboard = NSStoryboard(name: "Main", bundle: nil)
guard let controller:NSWindowController =
storyboard.instantiateControllerWithIdentifier("myWindowController") as? NSWindowController
else { return /*or handle error*/ }
controller.showWindow(self)
```

Ziel c

```
NSStoryboard *storyBoard = [NSStoryboard storyboardWithName:@"Main" bundle:nil]; // get a
reference to the storyboard
myController = [storyBoard instantiateControllerWithIdentifier:@"secondWindowController"]; //
instantiate your window controller
[myController showWindow:self];
```

Wenn Sie Ihren `controller` stellen Sie sicher, dass Sie einen Verweis darauf außerhalb des Funktionsaufrufs beibehalten. Dazu können Sie eine `NSWindowController` Variable in Ihrem App-Delegaten erstellen und den neuen Controller der Variablen zuweisen.

NSStoryBoard online lesen: https://riptutorial.com/de/osx/topic/4287/nsstoryboard

Kapitel 8: NSTextView

Einführung

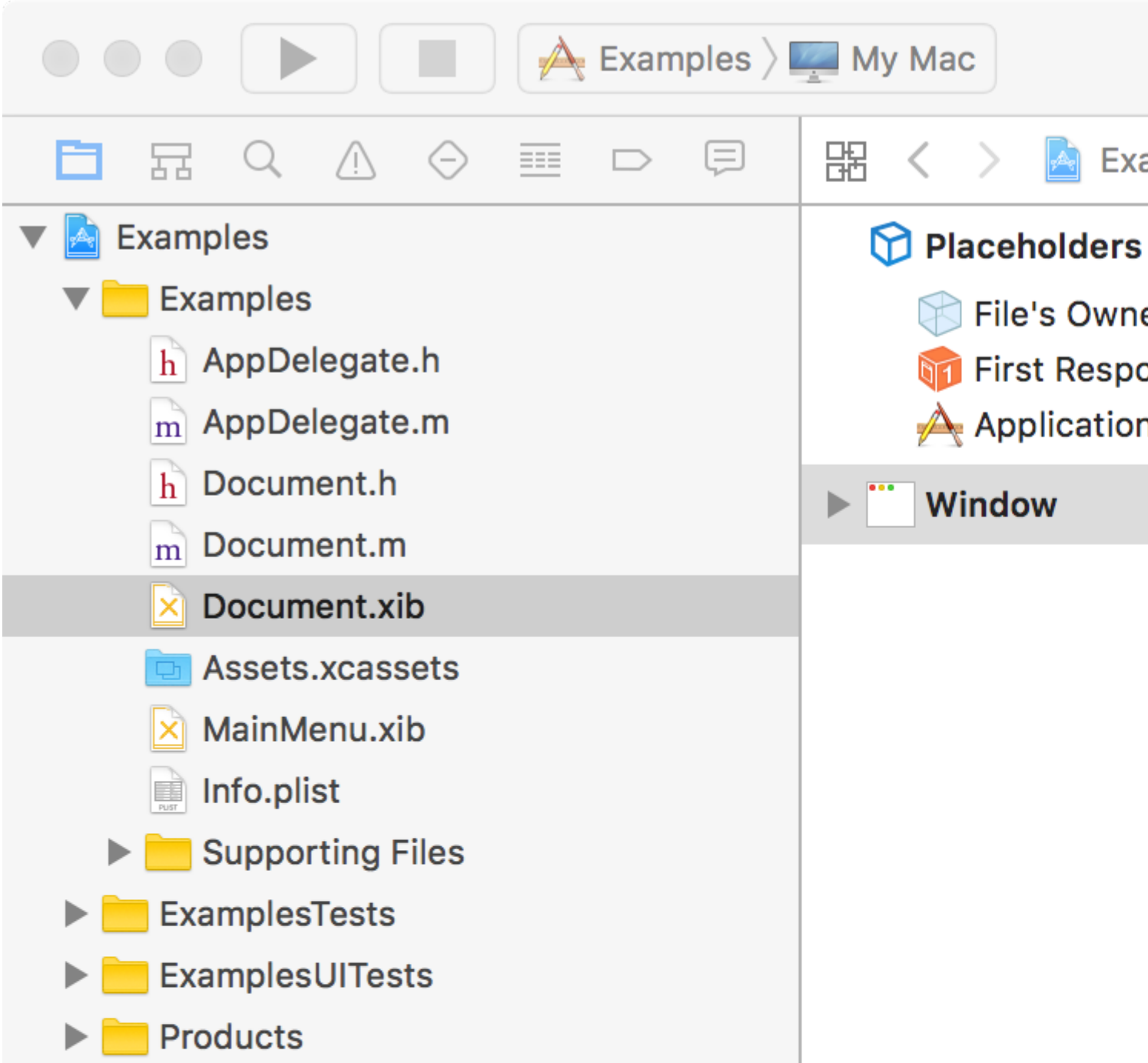
NSTextView ist Apples Haupthandler für das Textsystem von AppKit. Es enthält alles, was Sie zum Erstellen eines Text-Viewer / Editor-Bereichs in OS X-Anwendungen (umbenannte macOS-Anwendungen) benötigen.

Examples

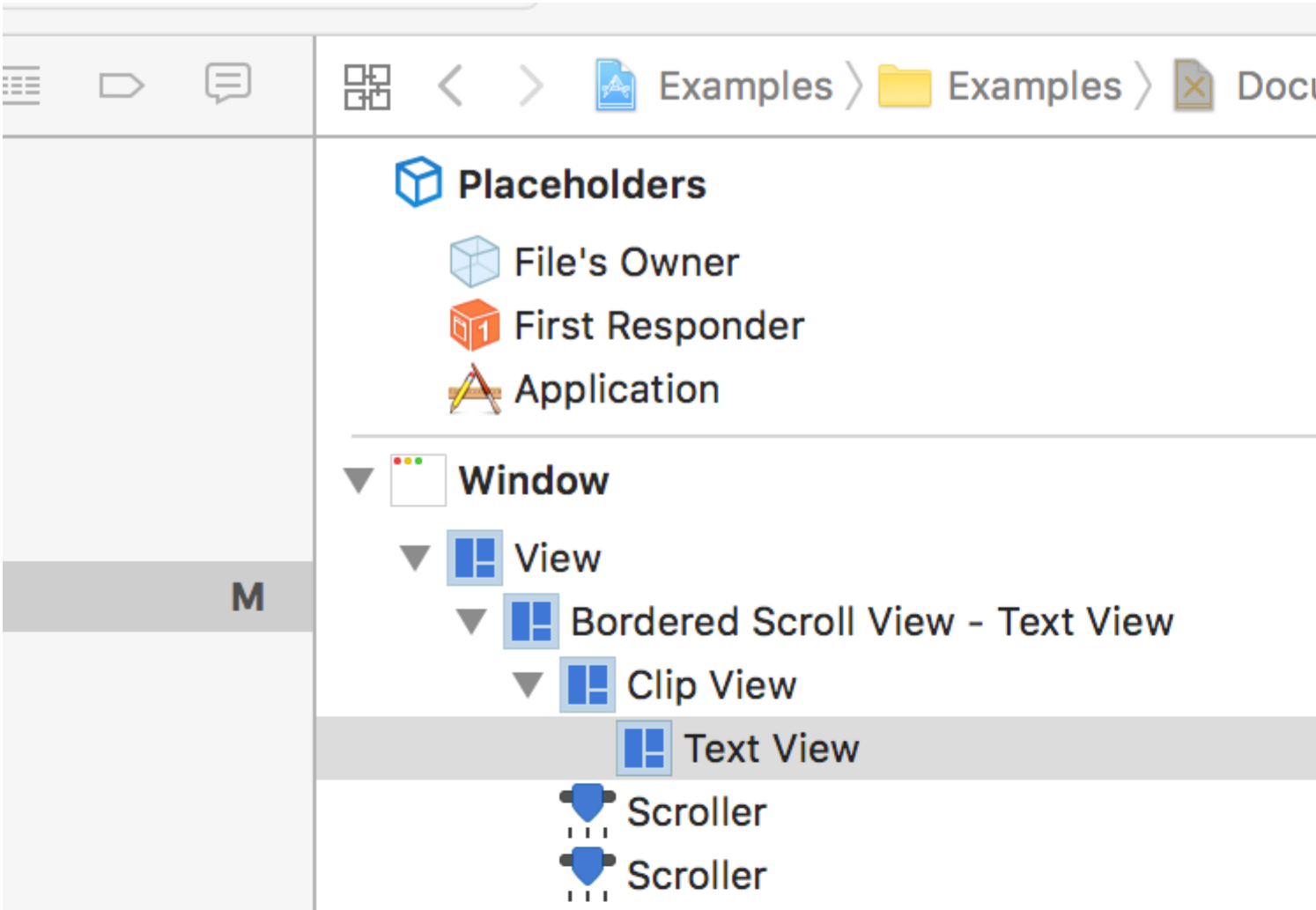
NSTextView erstellen

Grafisch

In XCode kann eine einfache NSTextView durch Ziehen und Ablegen aus der Objektbibliothek erstellt werden.


Examples
My Mac
Examples
Examples
AppDelegate.h
AppDelegate.m
Document.h
Document.m
Document.xib
Assets.xcassets
MainMenu.xib
Info.plist
Supporting Files
ExamplesTests
ExamplesUITests
Products
Placeholders
File's Owne
First Respo
Application
Window

, die automatisch so eingestellt wird, dass sie vertikal mit der **Textansicht** erweitert wird. Stellen Sie sicher, dass Sie beim Option () -Dragging Verbindungen zur Textansicht und nicht zur Bildlaufansicht herstellen.



Drag out to expose

1. Ein NSLayoutManager - führt ein Glyphen- / Zeichen-Layout aus.
2. Ein NSTextContainer - steuert den grafischen Bereich, den Glyphen / Zeichen enthalten können.
3. Ein NSTextStorage - enthält die tatsächlichen Zeichenfolgendaten, die NSTextView anzeigt.

- Ein NSTextStorage kann viele NSLayoutManager haben, ein NSLayoutManager kann jedoch nur einen NSTextStorage haben. Dies ist nützlich, wenn Sie dieselben Daten auf unterschiedliche Weise gleichzeitig anzeigen möchten.
- NSLayoutManager kann viele NSTextContainer haben. Nützlich für paginierten Text.
- NSTextView kann jeweils nur einen NSTextContainer haben.
- Bestimmte in NSTextView integrierte Dinge sind zum Zeitpunkt des Schreibens nicht zulässig. Beispielsweise können integrierte Such- und Ersetzungsfunktionen nicht angepasst werden, sie können jedoch mit benutzerdefinierten Funktionen überschrieben werden.

Weitere Informationen zur Verwendung des Textsystems finden Sie hier .

Nun zum Code. Mit diesem Code wird eine einfache NSTextView erstellt, bei der nicht einmal gescrollt wird. Solche Dinge wie Scrollen und Paginieren werden in einem anderen Beispiel sein.

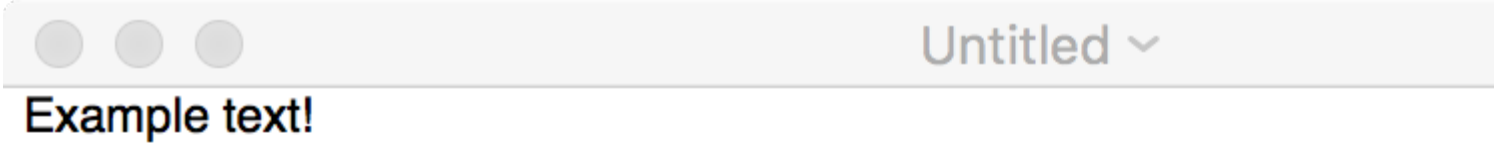
Ziel c

```
// This code resides in an NSDocument object's windowControllerDidLoadNib:(NSWindowController
*)windowController method.
// This is done simply because it is easy and automatically gets called upon.

// This method is also where the following NSRect variable gets size information. We need this
information for this example.
NSRect windowFrame = windowController.window.contentView.frame;
NSTextStorage *textStorage = [[NSTextStorage alloc] initWithString:@"Example text!"];
NSLayoutManager *manager = [[NSLayoutManager alloc] init];
NSTextContainer *container = [[NSTextContainer alloc]
initWithContainerSize:NSMakeSize(windowFrame.size.width, windowFrame.size.height)];
NSTextView *textView = [[NSTextView alloc] initWithFrame:windowFrame textContainer:container];

[textStorage addLayoutManager:manager];
[manager addTextContainer:container];
[windowController.window setContentView:textView];
```

Herzliche Glückwünsche! Sie haben programmgesteuert eine NSTextView erstellt!



NSTextView online lesen: https://riptutorial.com/de/osx/topic/8880/nstextview

Kapitel 9: Umgebungsvariablen setzen

Einführung

In Mac OS X können Sie die Umgebungsvariablen in einer der folgenden Dateien festlegen:

~ / .bashrc

~ / .bash_profile

~ / .profile

Standardmäßig enthält Mac OS X keine Dateien, die Sie manuell erstellen müssen.

Examples

Pfad hinzufügen

1. `vim ~/.bash_profile`

Die Datei ist möglicherweise nicht vorhanden (andernfalls können Sie sie einfach erstellen).

2. Geben Sie hier ein und speichern Sie die Datei:

```
export PATH=$PATH:YOUR_PATH_HERE
```

Umgebungsvariablen setzen online lesen:
https://riptutorial.com/de/osx/topic/10162/umgebungsvariablen-setzen

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit osx	Andrew Hoos, Community, l'L'l, tbodt
2	Dateizuordnung	bikram990
3	Fordern Sie den Benutzer zur Eingabe einer Datei auf	Andrew Hoos
4	NSFont	malicedShade
5	NSMenuItem	Andrew Hoos, Barlow Tucker
6	NSRunLoop	Marco Pashkov
7	NSStoryBoard	Barlow Tucker
8	NSTextView	malicedShade
9	Umgebungsvariablen setzen	Kuhan