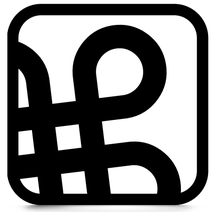


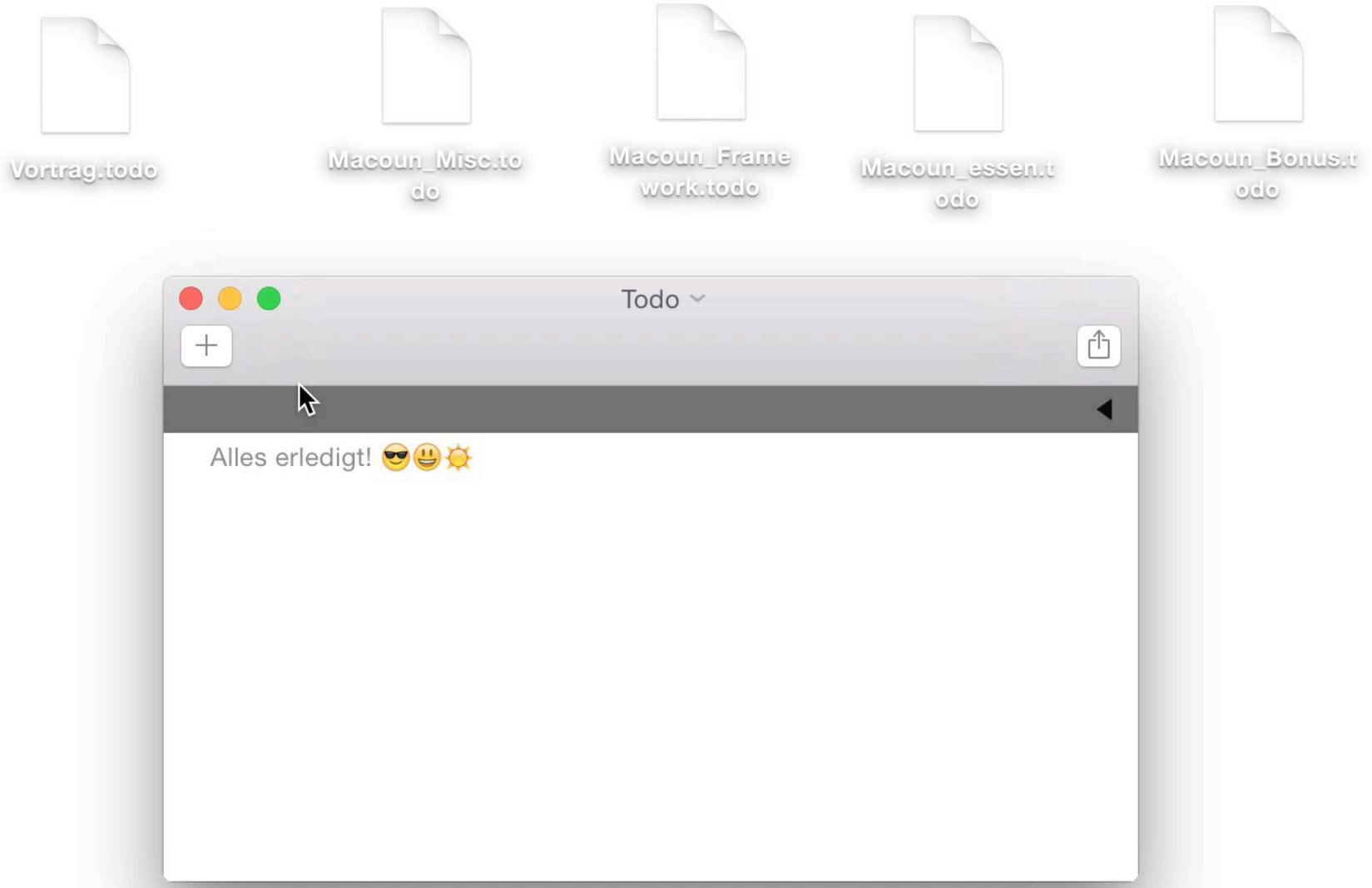
Macoun

Cocoa Apps für 10.10

(oder: iOS-Entwickler beginnen Mac-Entwicklung zu verstehen)

Florian Bachmann
Deutsche Telekom AG

stark inspiriert
von den
WWDC14
Sessions 204, 212 & 227

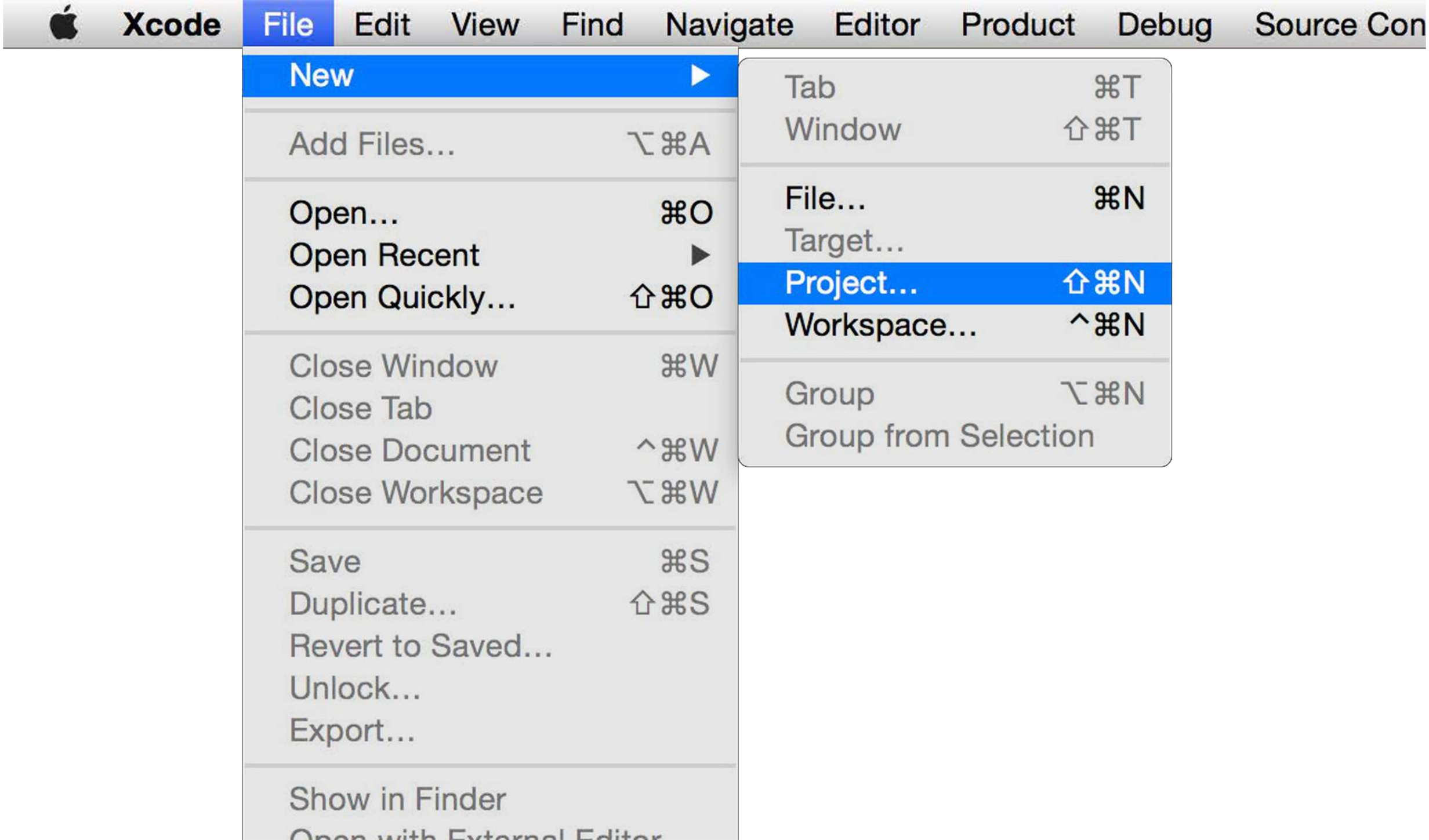


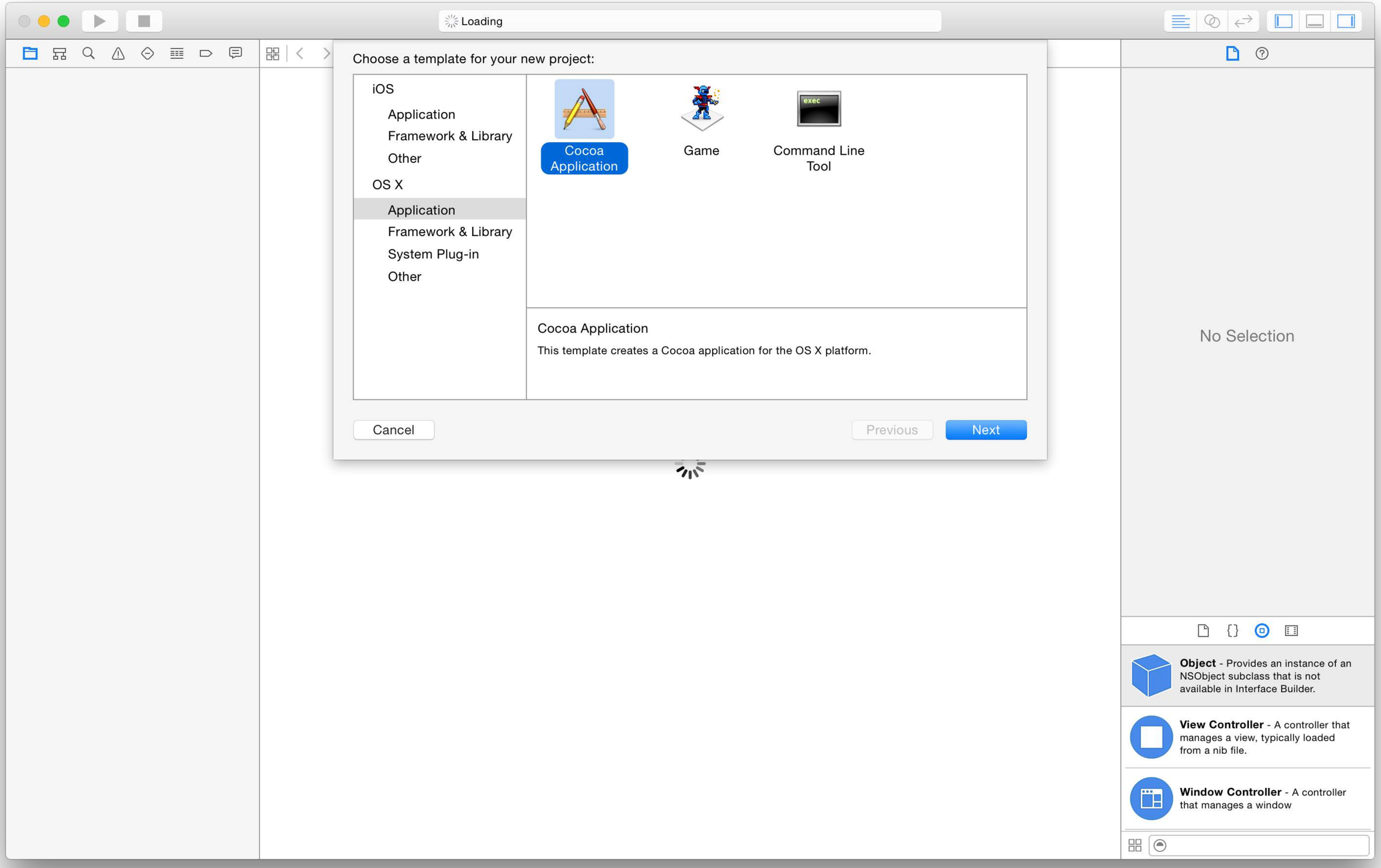
Macoun Todo

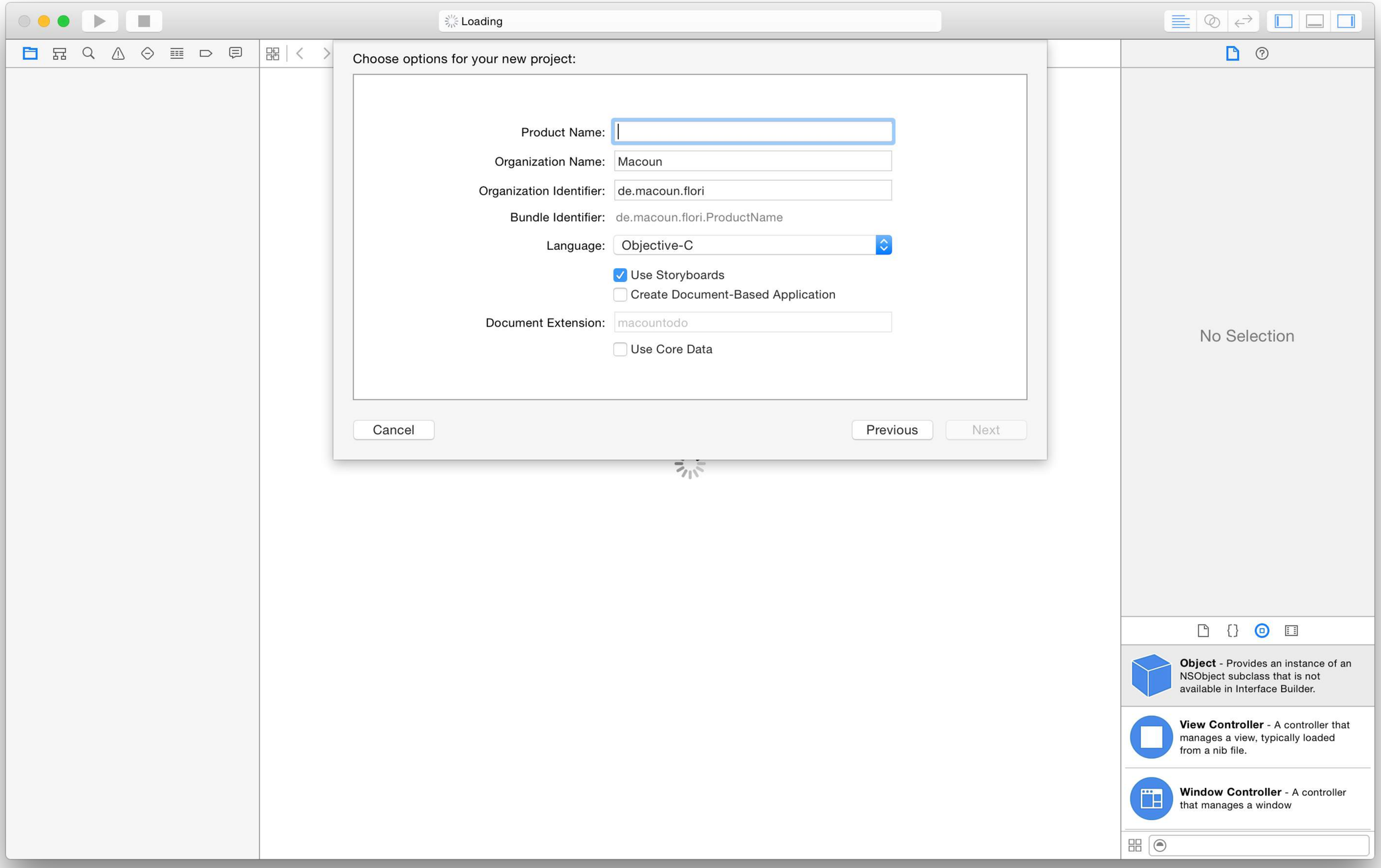
- OS X
- Storyboards & Autolayout
- Frameworks
- NSDocument

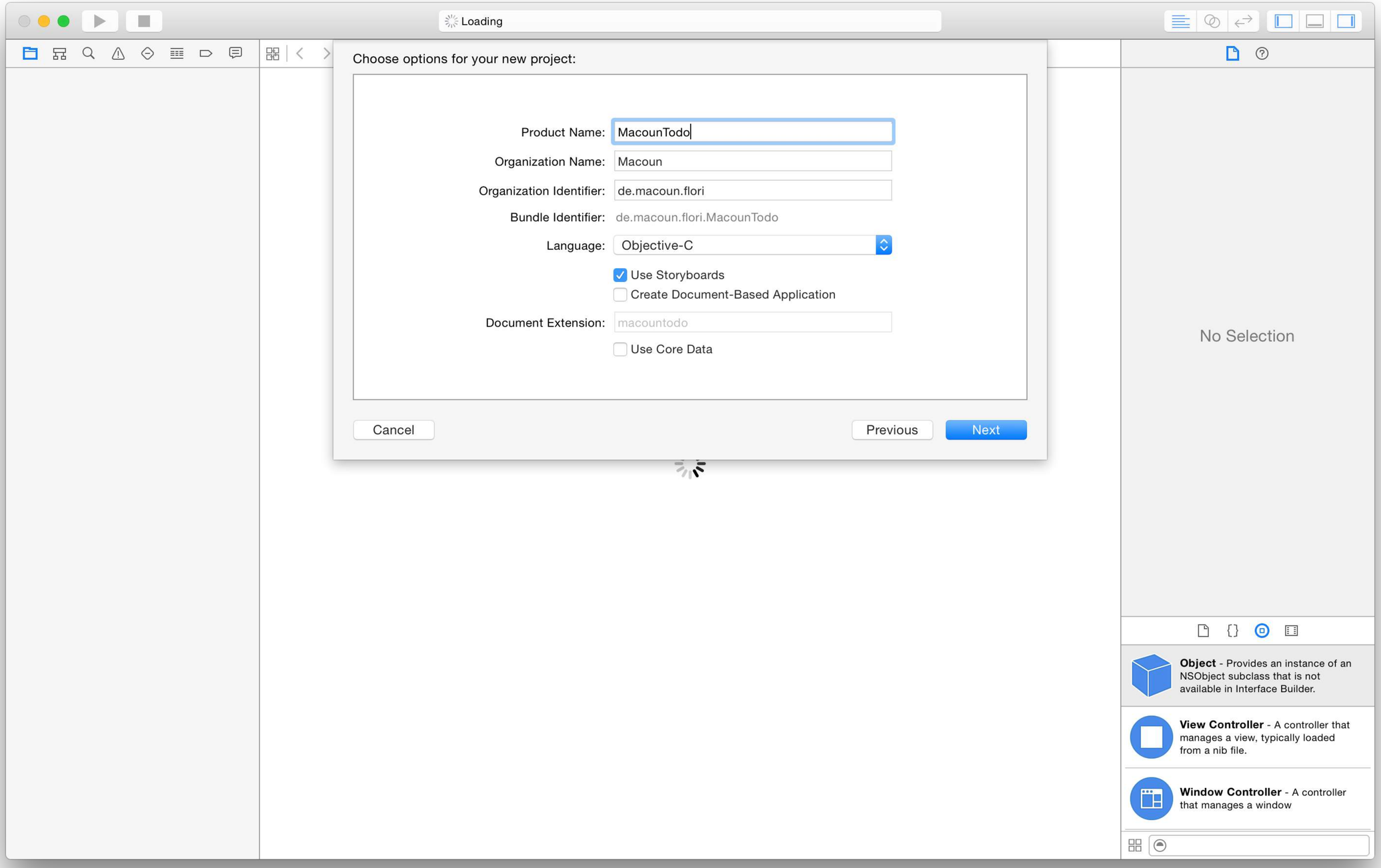
“Lister” nachprogrammieren

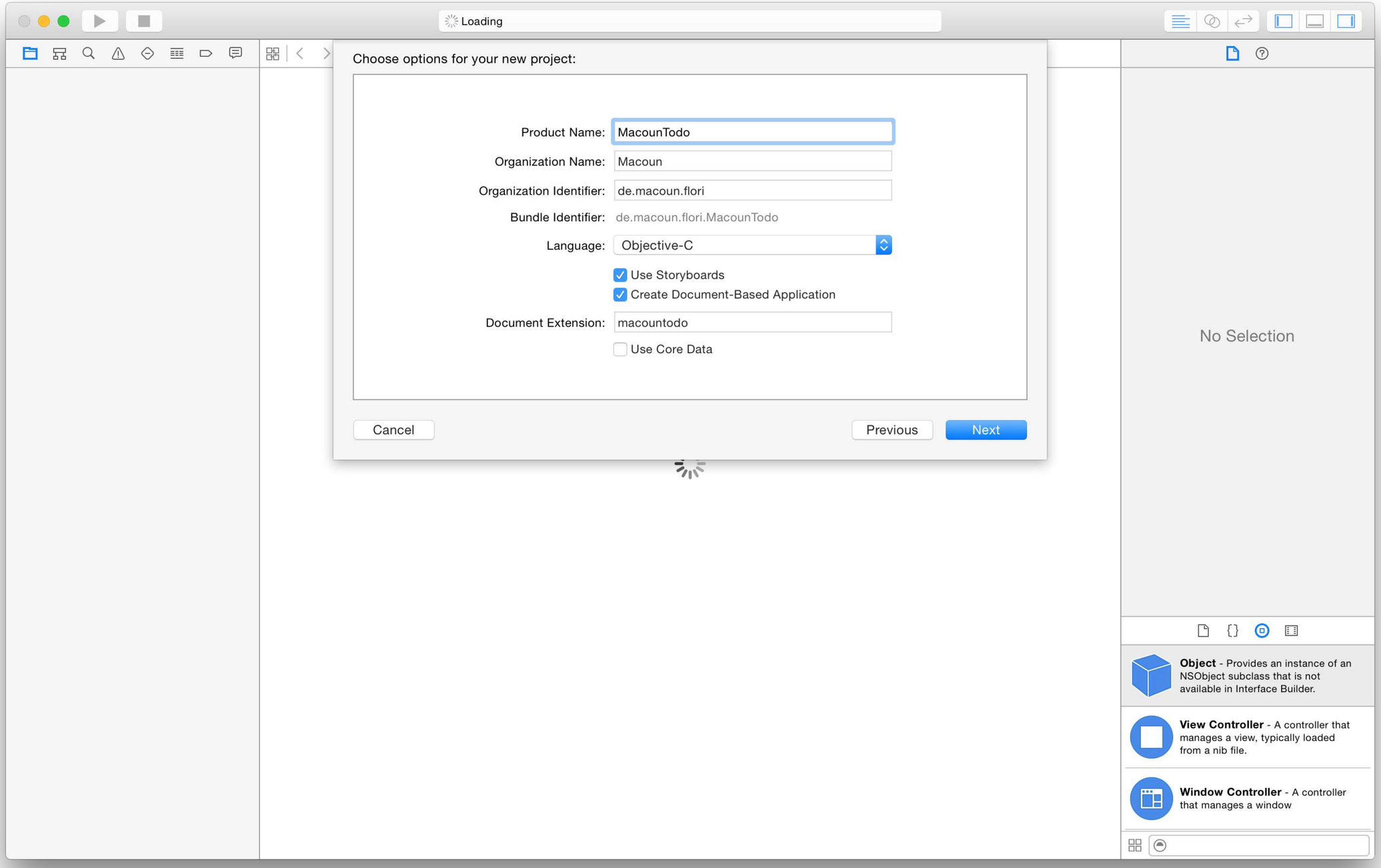
(204, 212 & 227)

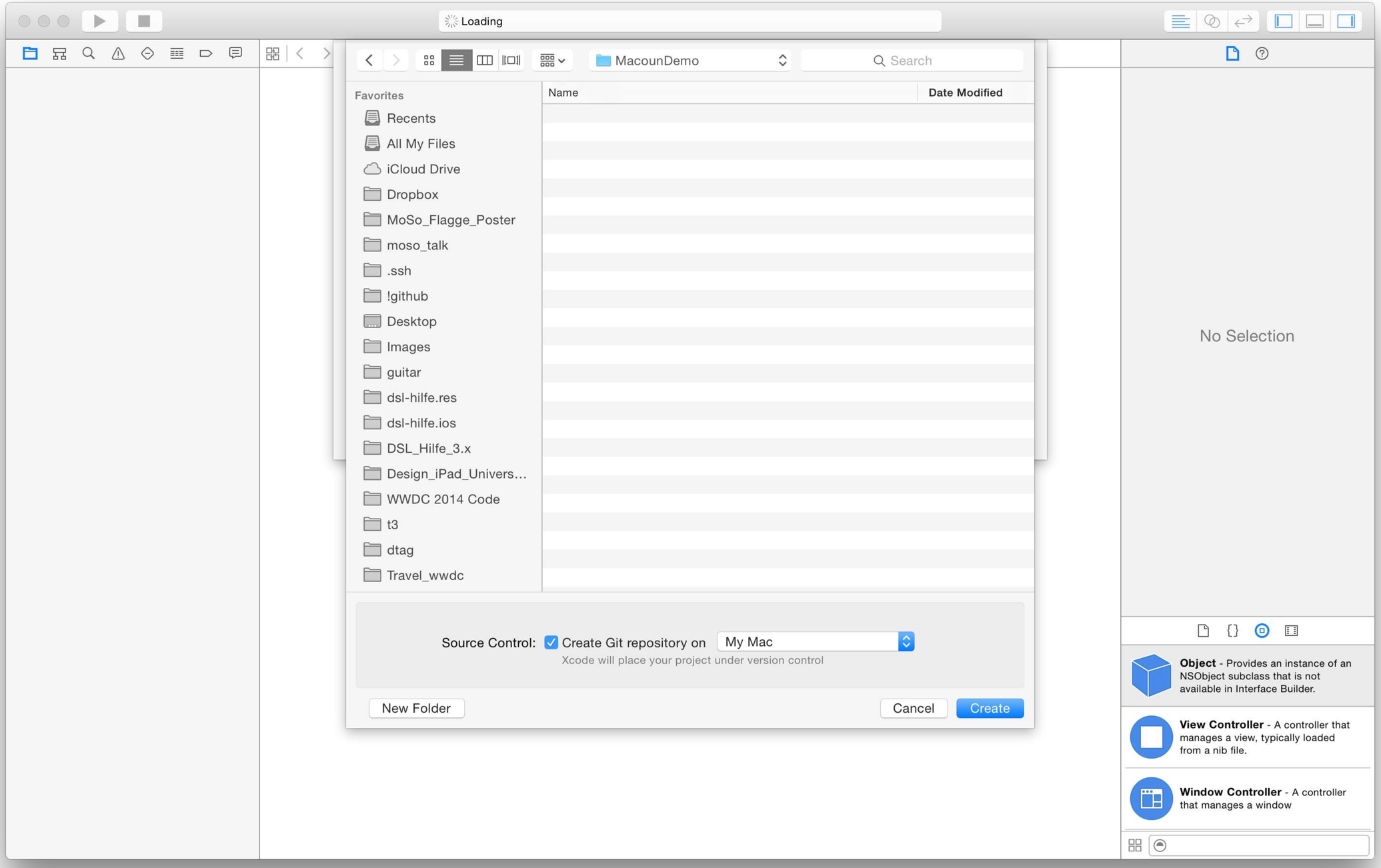

Xcode
File
Edit
View
Find
Navigate
Editor
Product
Debug
Source Con
New
Add Files… ⌥⌘A
Open… ⌘O
Open Recent
Open Quickly… ⇧⌘O
Close Window ⌘W
Close Tab
Close Document ^⌘W
Close Workspace ⌥⌘W
Save ⌘S
Duplicate… ⇧⌘S
Revert to Saved…
Unlock…
Export…
Show in Finder
Tab ⌘T
Window ⇧⌘T
File… ⌘N
Target…
Project… ⇧⌘N
Workspace… ^⌘N
Group ⌥⌘N
Group from Selection

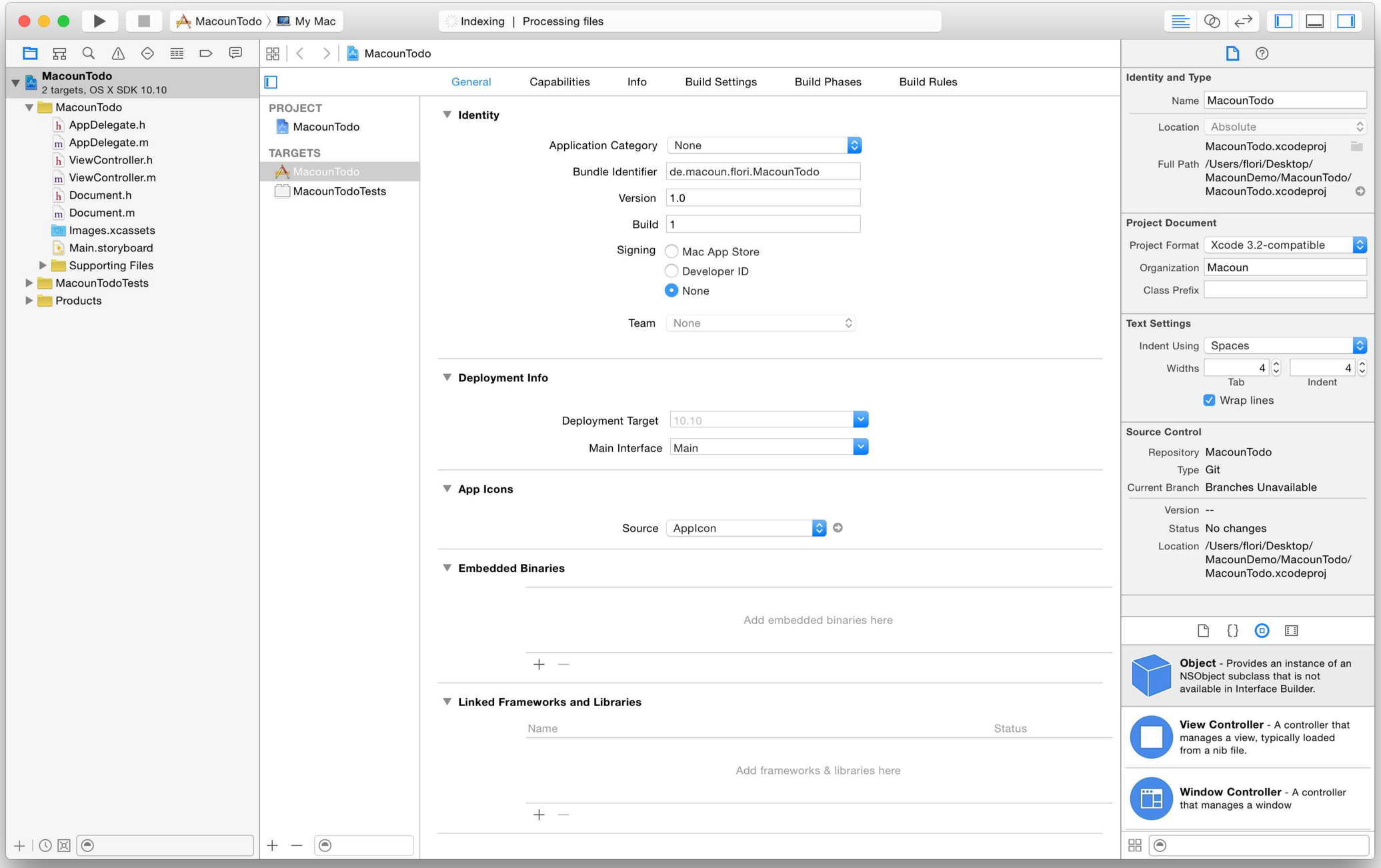

Loading
Choose a template for your new project:
iOS
Application
Framework & Library
Other
OS X
Application
Framework & Library
System Plug-in
Other
Cocoa Application
Game
exec
Command Line Tool
Cocoa Application
This template creates a Cocoa application for the OS X platform.
Cancel
Previous
Next
No Selection
Object - Provides an instance of an NSObject subclass that is not available in Interface Builder.
View Controller - A controller that manages a view, typically loaded from a nib file.
Window Controller - A controller that manages a window

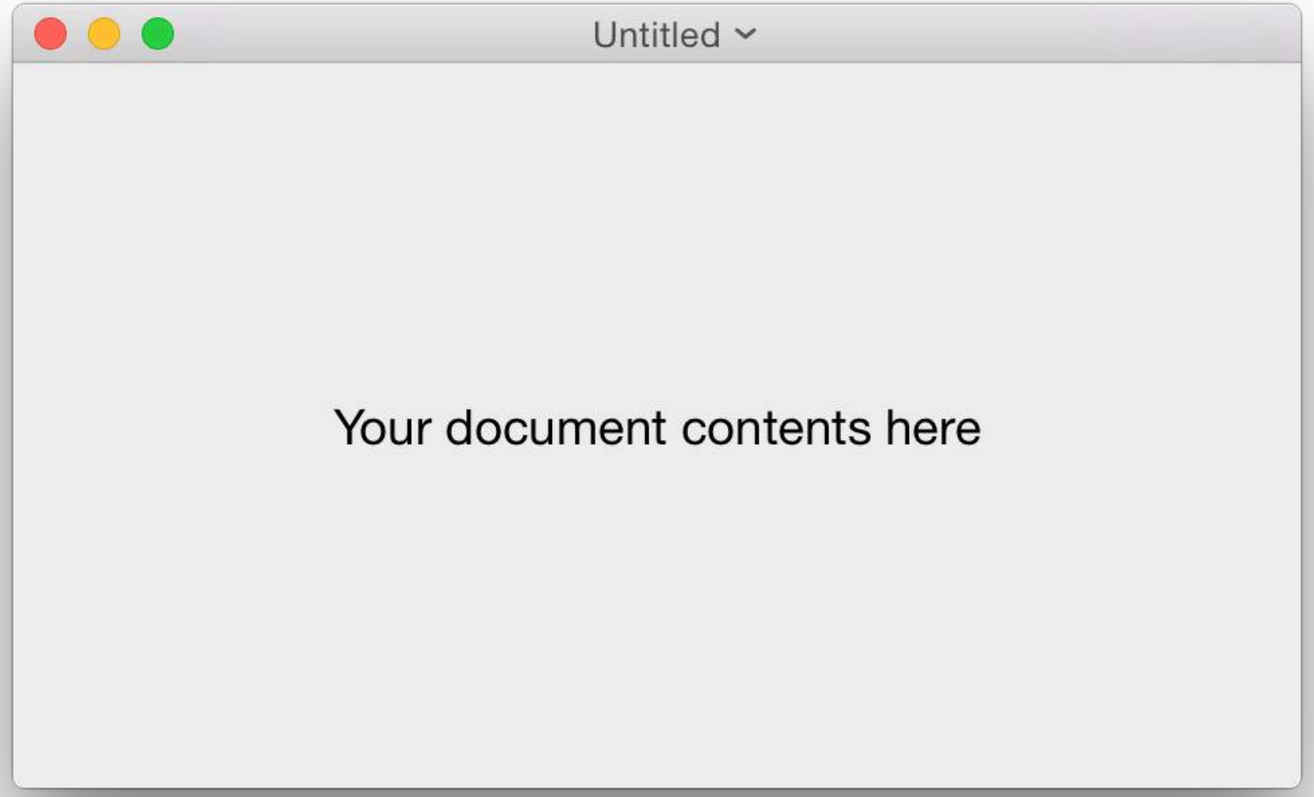

Loading
Choose options for your new project:
Product Name:
Organization Name:
Macoun
Organization Identifier:
de.macoun.flori
Bundle Identifier:
de.macoun.flori.ProductName
Language:
Objective-C
Use Storyboards
Create Document-Based Application
Document Extension:
macountodo
Use Core Data
Cancel
Previous
Next
No Selection
Object - Provides an instance of an NSObject subclass that is not available in Interface Builder.
View Controller - A controller that manages a view, typically loaded from a nib file.
Window Controller - A controller that manages a window


Loading
Choose options for your new project:
Product Name: MacounTodo
Organization Name: Macoun
Organization Identifier: de.macoun.flori
Bundle Identifier: de.macoun.flori.MacounTodo
Language: Objective-C
Use Storyboards
Create Document-Based Application
Document Extension: macountodo
Use Core Data
Cancel
Previous
Next
No Selection
Object - Provides an instance of an NSObject subclass that is not available in Interface Builder.
View Controller - A controller that manages a view, typically loaded from a nib file.
Window Controller - A controller that manages a window


Loading
Choose options for your new project:
Product Name: MacounTodo
Organization Name: Macoun
Organization Identifier: de.macoun.flori
Bundle Identifier: de.macoun.flori.MacounTodo
Language: Objective-C
Use Storyboards
Create Document-Based Application
Document Extension: macountodo
Use Core Data
Cancel
Previous
Next
No Selection
Object - Provides an instance of an NSObject subclass that is not available in Interface Builder.
View Controller - A controller that manages a view, typically loaded from a nib file.
Window Controller - A controller that manages a window


Loading
MacounDemo
Search
Favorites
Recents
All My Files
iCloud Drive
Dropbox
MoSo_Flagge_Poster
moso_talk
.ssh
!github
Desktop
Images
guitar
dsl-hilfe.res
dsl-hilfe.ios
DSL_Hilfe_3.x
Design_iPad_Univers…
WWDC 2014 Code
t3
dtag
Travel_wwdc
Name
Date Modified
Source Control:
Create Git repository on
My Mac
Xcode will place your project under version control
New Folder
Cancel
Create
No Selection
Object - Provides an instance of an NSObject subclass that is not available in Interface Builder.
View Controller - A controller that manages a view, typically loaded from a nib file.
Window Controller - A controller that manages a window


MacounTodo
My Mac
Indexing | Processing files
MacounTodo
2 targets, OS X SDK 10.10
MacounTodo
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
Document.h
Document.m
Images.xcassets
Main.storyboard
Supporting Files
MacounTodoTests
Products
General
Capabilities
Info
Build Settings
Build Phases
Build Rules
PROJECT
MacounTodo
TARGETS
MacounTodo
MacounTodoTests
Identity
Application Category
None
Bundle Identifier
de.macoun.flori.MacounTodo
Version
1.0
Build
1
Signing
Mac App Store
Developer ID
None
Team
None
Deployment Info
Deployment Target
10.10
Main Interface
Main
App Icons
Source
AppIcon
Embedded Binaries
Add embedded binaries here
Linked Frameworks and Libraries
Name
Status
Add frameworks & libraries here
Identity and Type
Name
MacounTodo
Location
Absolute
MacounTodo.xcodeproj
Full Path
/Users/flori/Desktop/MacounDemo/MacounTodo/MacounTodo.xcodeproj
Project Document
Project Format
Xcode 3.2-compatible
Organization
Macoun
Class Prefix
Text Settings
Indent Using
Spaces
Widths
4
Tab
4
Indent
Wrap lines
Source Control
Repository
MacounTodo
Type
Git
Current Branch
Branches Unavailable
Version
--
Status
No changes
Location
/Users/flori/Desktop/MacounDemo/MacounTodo/MacounTodo.xcodeproj
Object - Provides an instance of an NSObject subclass that is not available in Interface Builder.
View Controller - A controller that manages a view, typically loaded from a nib file.
Window Controller - A controller that manages a window


Untitled
Your document contents here

Basis-Klassen

`NSWindowController`

- An `NSWindowController` object manages a window

~~`UI`~~`NSViewController` (ganz viele Spezialisierungen: Tab, Split, …)

- A view controller manages a view (hängen nun in der *ResponderChain*)

~~`UI`~~`NSView`

- The `NSView` class defines the basic drawing, event-handling, and printing architecture of an app.

NSDocument

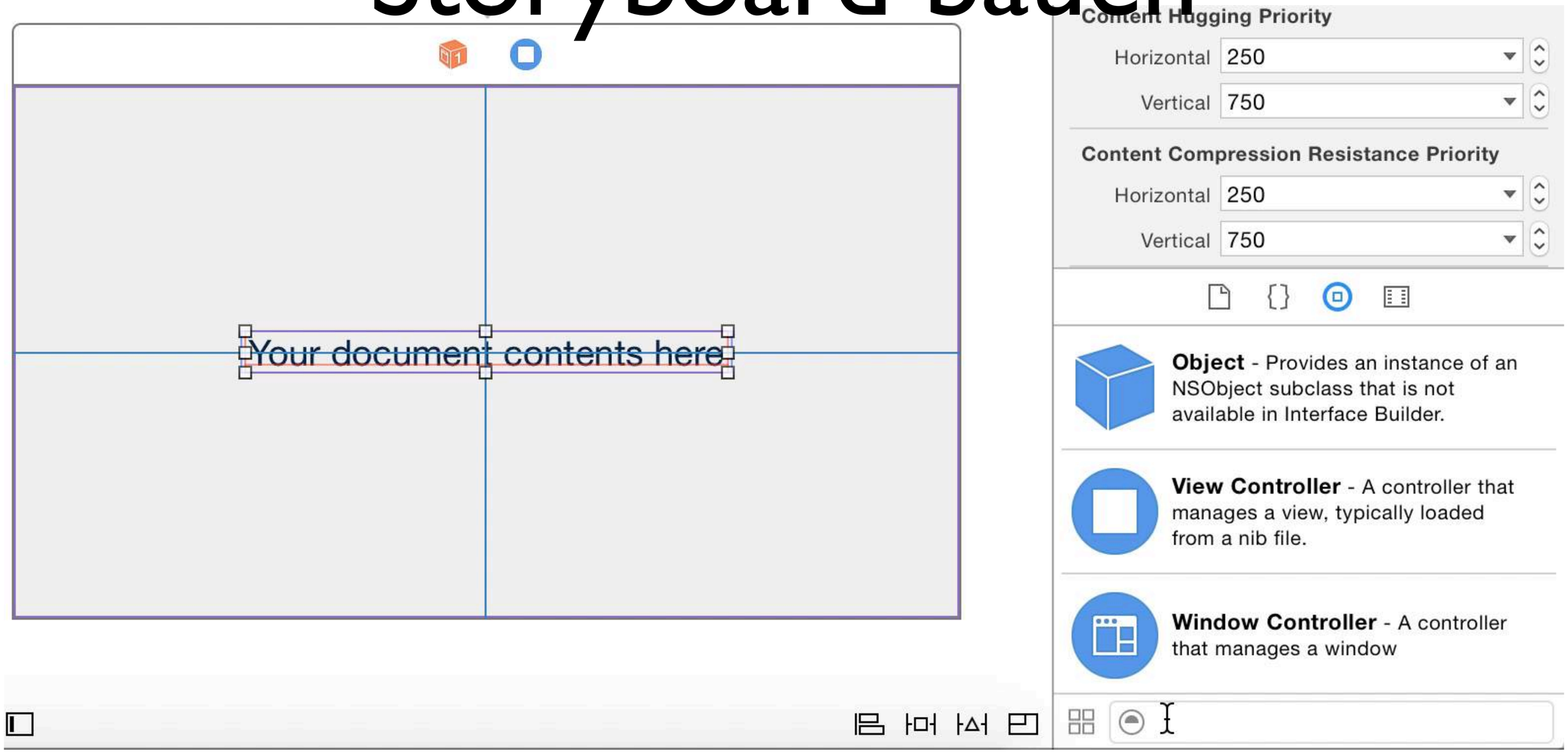
bietet schon viel von Haus aus an:

- Auto Save & Versions
- iCloud
- asynchrones lesen und schreiben
- Undo

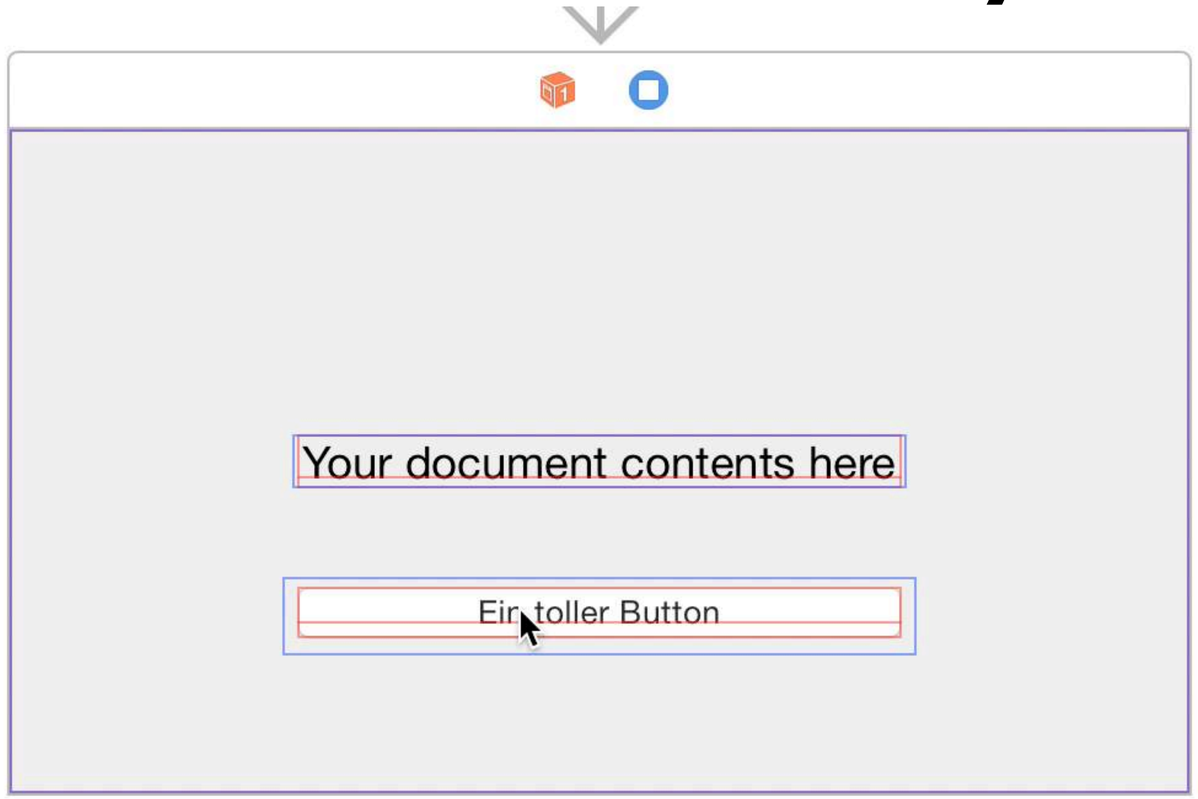
Storyboard

- Verbindung zwischen Model und View
- Alle Views platzieren (Text, Buttons, …)
- Auto Layout setzen
- IBActions und IBOutlets verbinden

Storyboard bauen



Storyboard bauen




```
3  //  MacounTodo
4  //
5
6  #import "ViewController.h"
7
8  @interface ViewController()
9  @property (nonatomic, strong) IBOutlet NSButton *tollerButton;
10 @end
11
12 @implementation ViewController
13
14 - (IBAction)wichtigerButtonAction:(NSButton*)sender {
15     NSLog(@"toll, du hast einen Button gedrückt: %@", sender.title);
16 }
17
18 @end
19
```

Storyboard - Segues

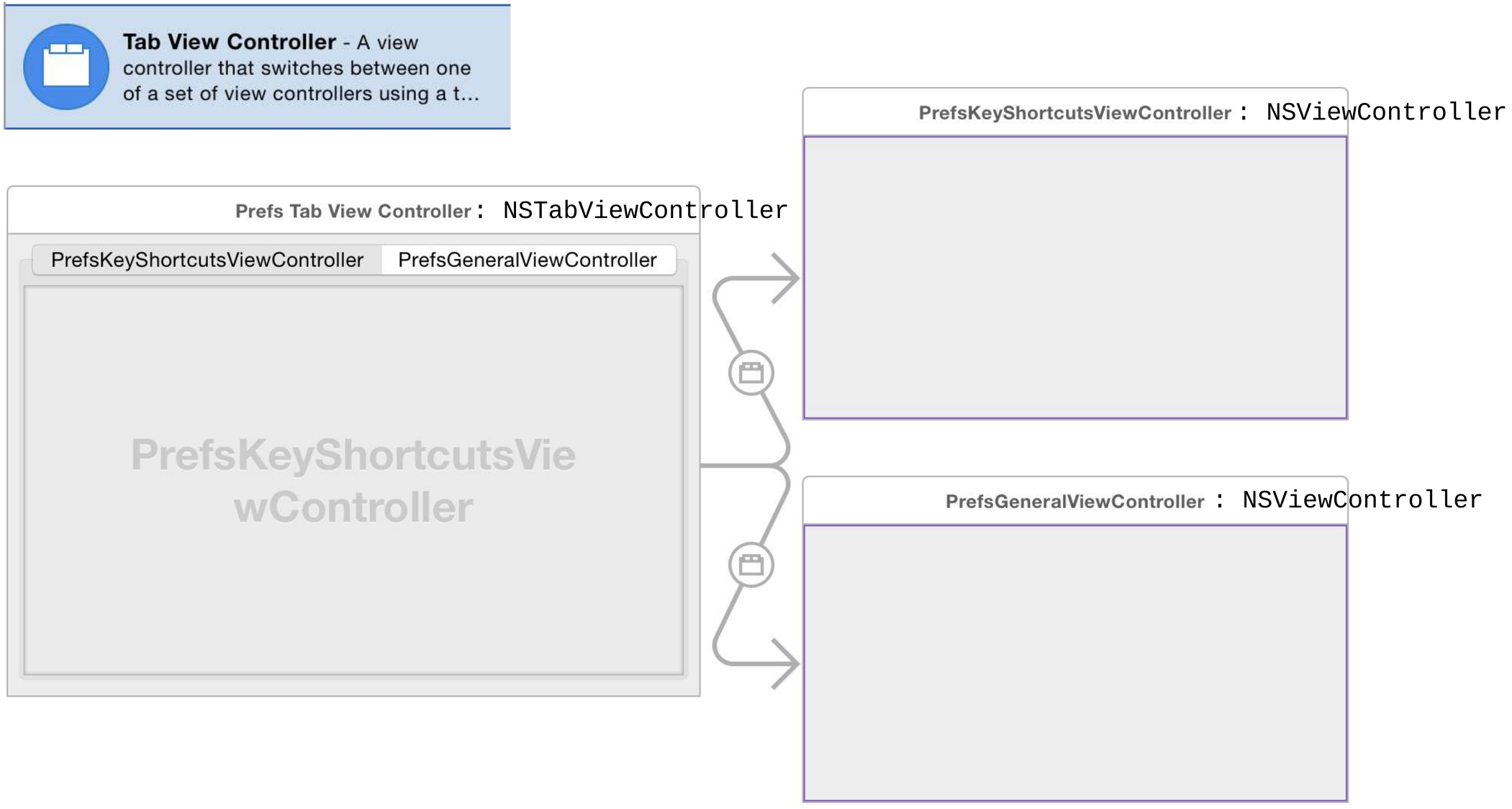
Containment

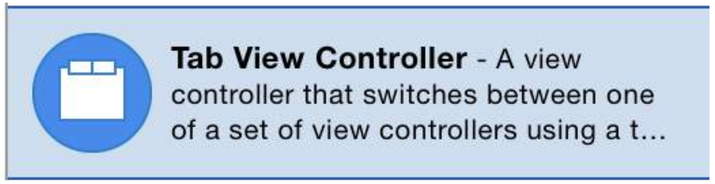
- Window
- Split
- Tab

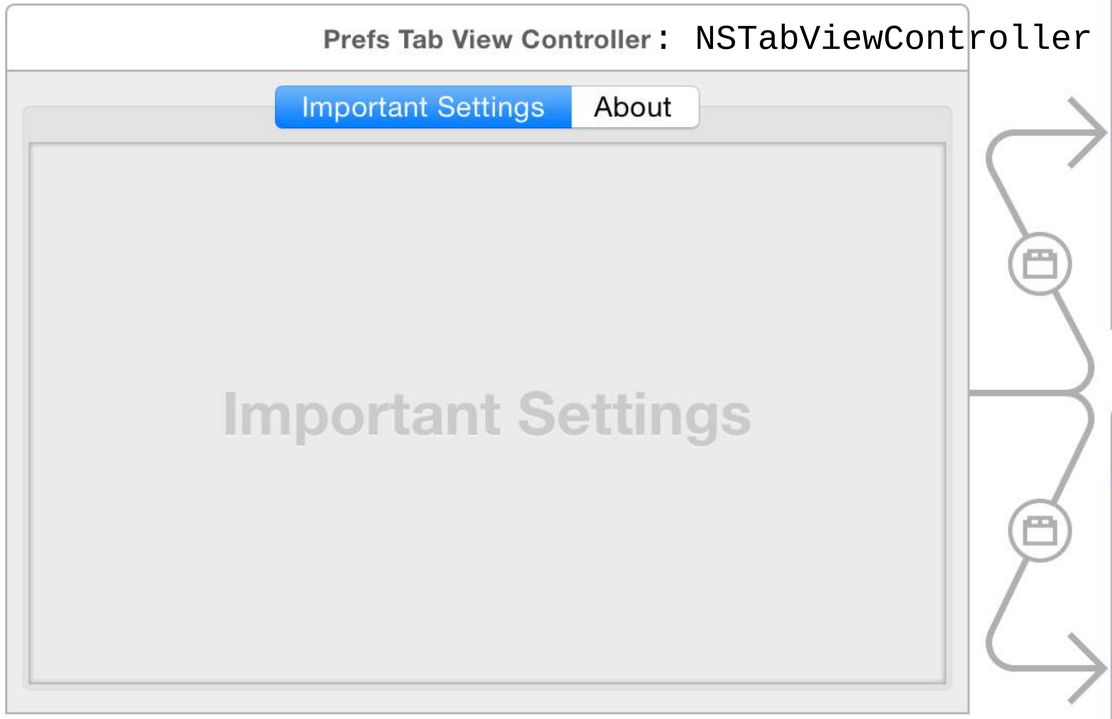
Presentation

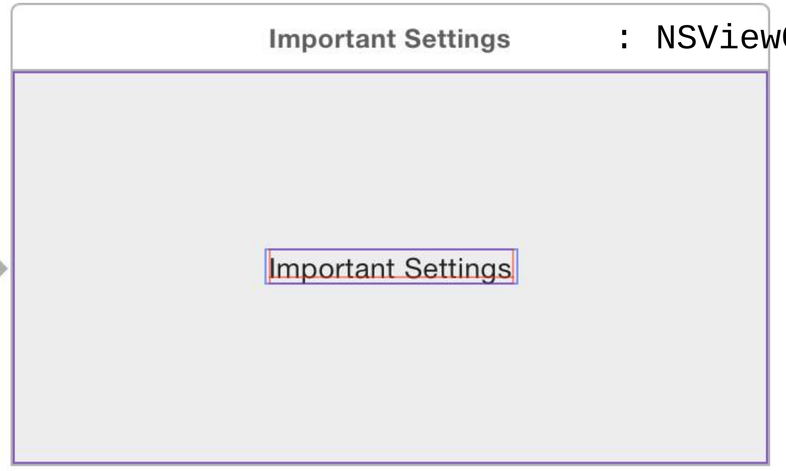
- Modal
- Sheet
- Popover

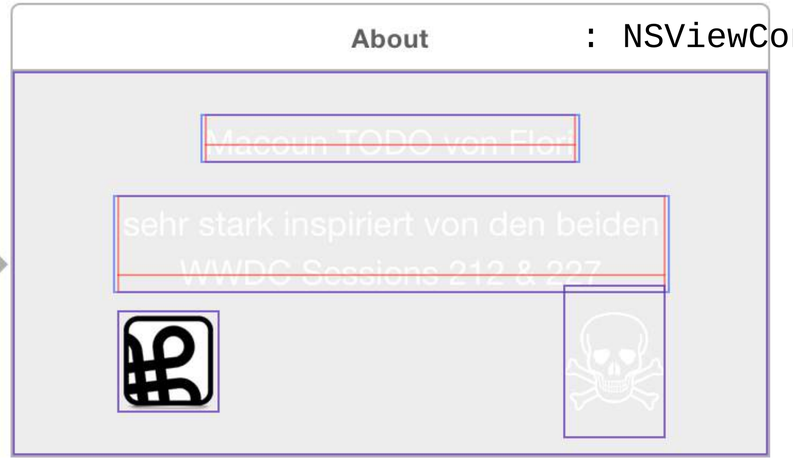
```
- (void)prepareForSegue:(NSStoryboardSegue *)segue
                 sender:(id)sender;
```


Tab View Controller - A view controller that switches between one of a set of view controllers using a t...
Prefs Tab View Controller : NSTabViewController
PrefsKeyShortcutsViewController
PrefsGeneralViewController
PrefsKeyShortcutsViewController
PrefsKeyShortcutsViewController : NSViewController
PrefsGeneralViewController
PrefsGeneralViewController : NSViewController


Tab View Controller - A view controller that switches between one of a set of view controllers using a t...


Prefs Tab View Controller : NSTabViewController
Important Settings
About
Important Settings


Important Settings : NSViewController
Important Settings


About : NSViewController
Maccun TODO von Flori
sehr stark inspiriert von den beiden
WWDC Sessions 212 & 227

View Controller

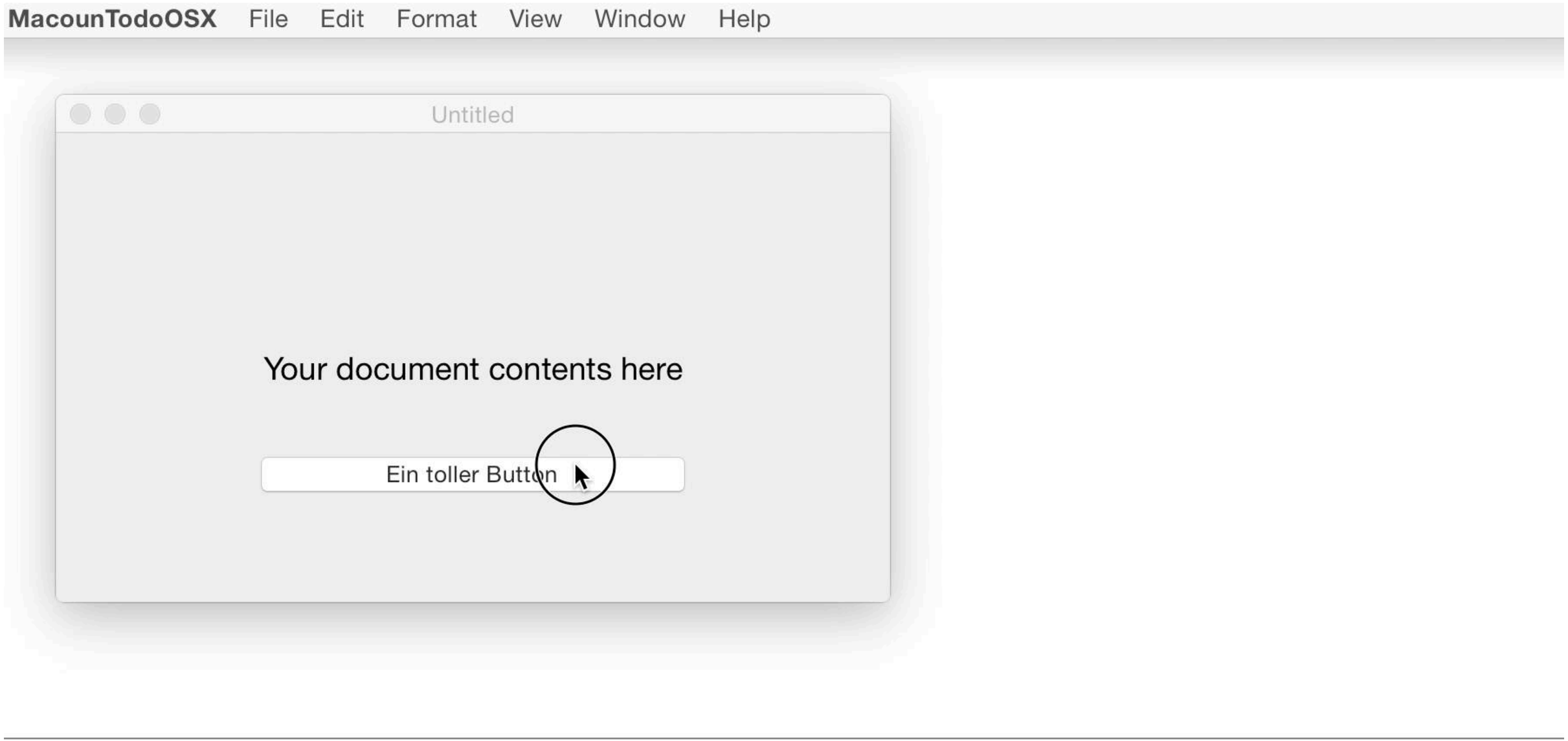
Your document contents here

Ein toller Button

Prefs Tab View Controller

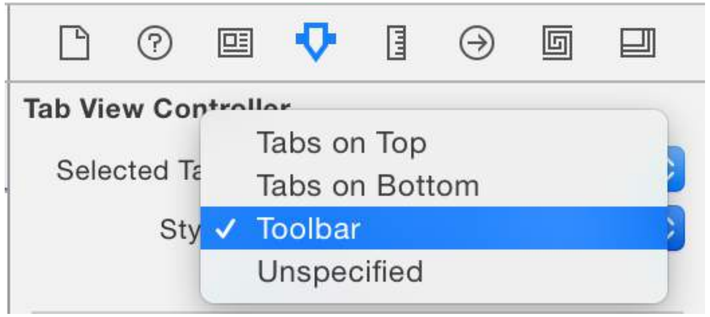
Important Settings
About

Important Settings

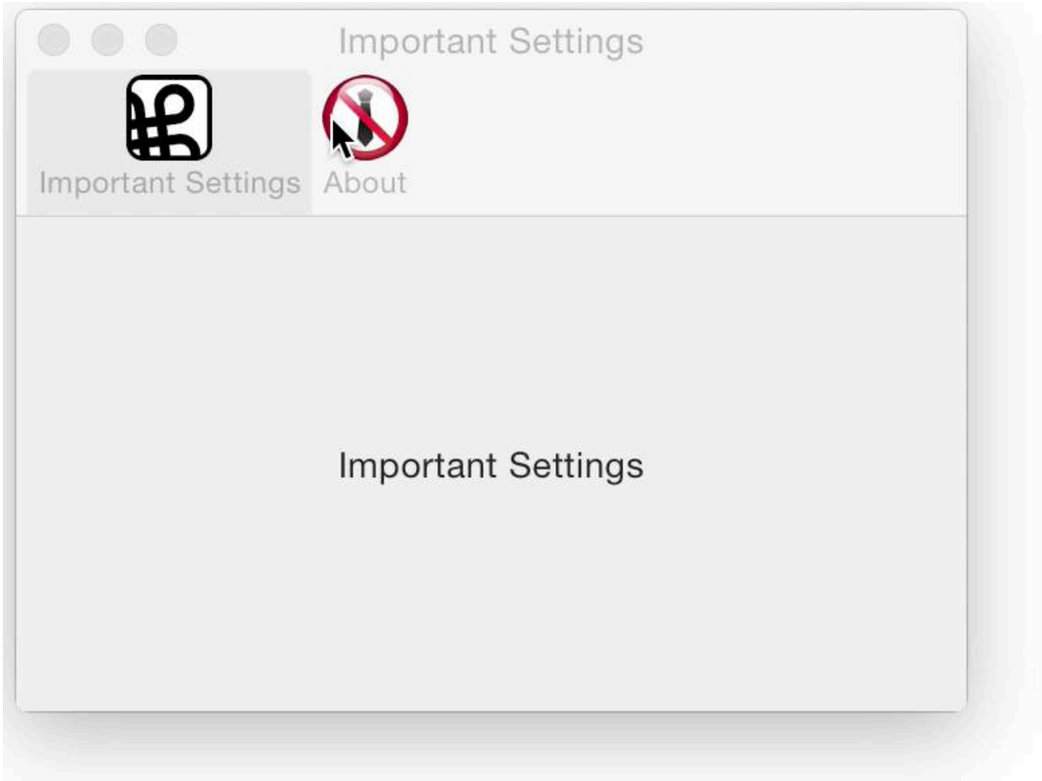

MacounTodoOSX File Edit Format View Window Help
Untitled
Your document contents here
Ein toller Button

1. Bilder reinkopieren

2. Toolbar auswählen

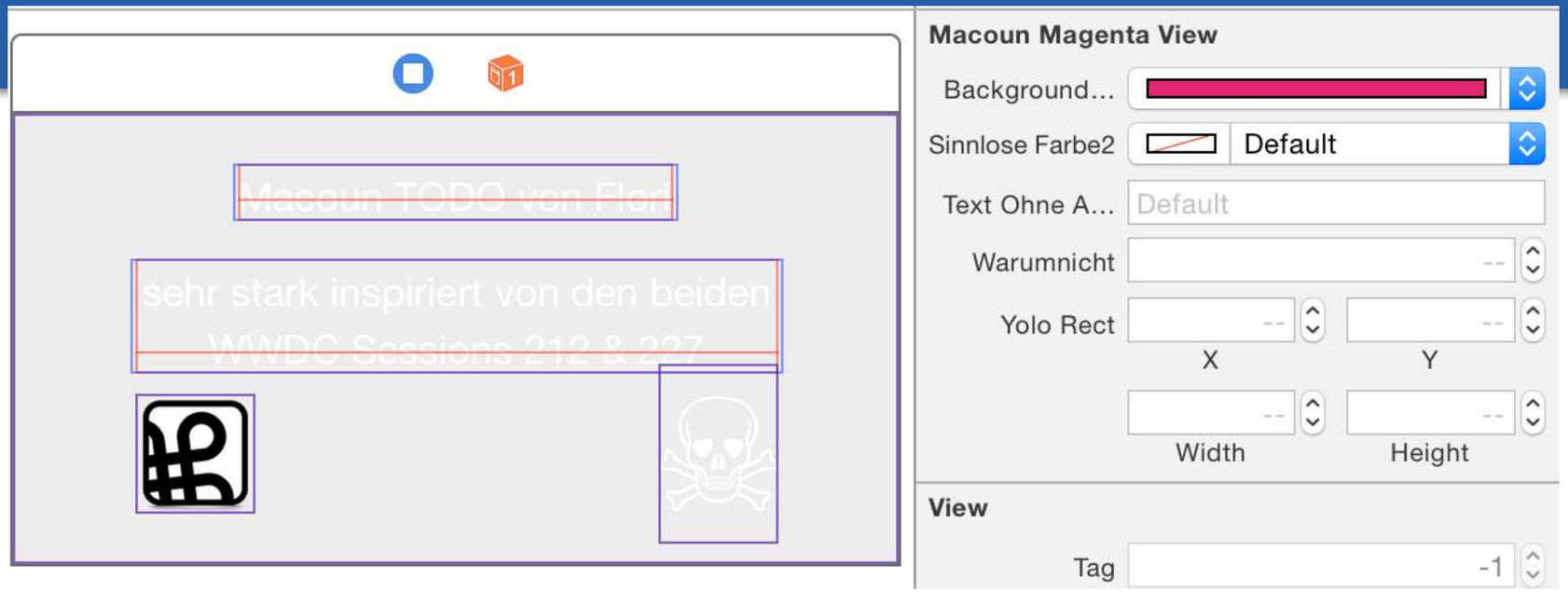


3. Like a pr0!



IB_DESIGNABLE

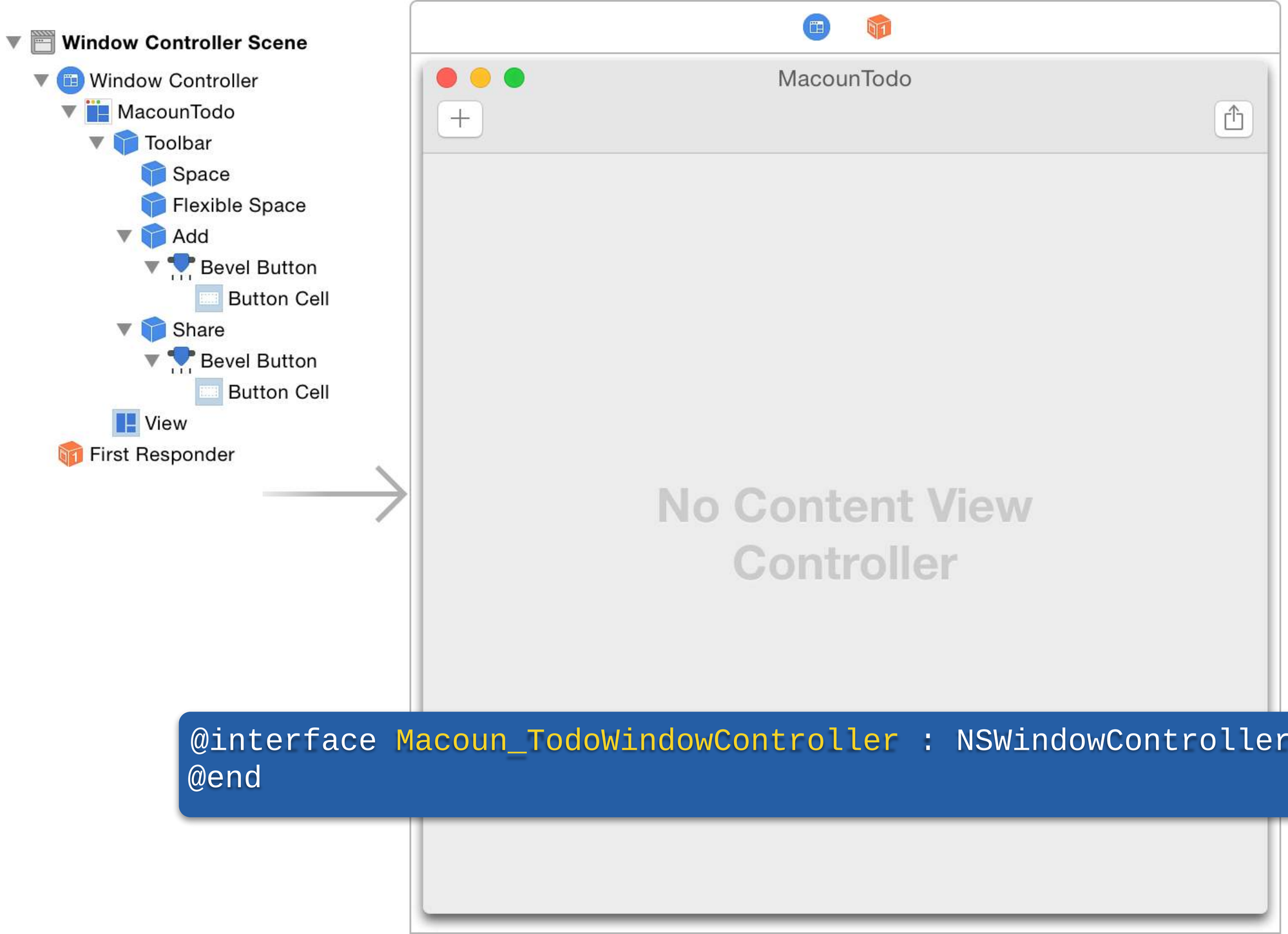
```
IB_DESIGNABLE
@interface Macoun_MagentaView : NSView
@property(nonatomic,strong) IBInspectable NSColor* backgroundColor;
@property(nonatomic,strong) IBInspectable NSColor* sinnloseFarbe2;
@property(nonatomic, copy)  IBInspectable NSString* textOhneAuswirkung;
@property(nonatomic)        IBInspectable CGFloat warumNicht;
@property(nonatomic)        IBInspectable NSRect yoloRect;
@end
```



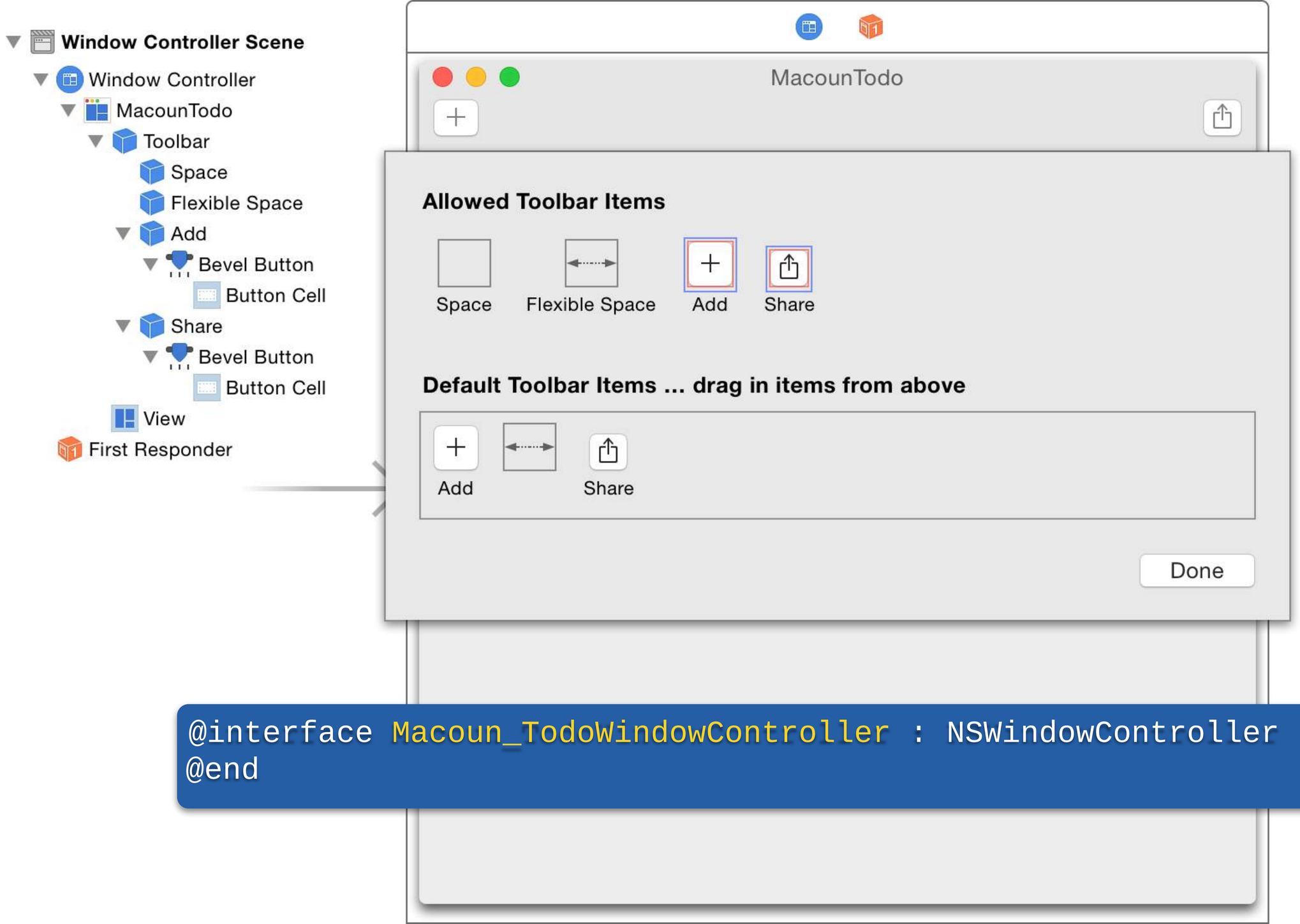
IB_DESIGNABLE

```
@implementation Macoun_MagentaView
- (void)awakeFromNib {
    if (!self.backgroundColor) {
        self.backgroundColor = [NSColor colorTelekom];
    }
}

- (void)drawRect:(NSRect)dirtyRect {
    [super drawRect:dirtyRect];
    [self.backgroundColor set];
    NSRectFill(dirtyRect);
}
@end
```

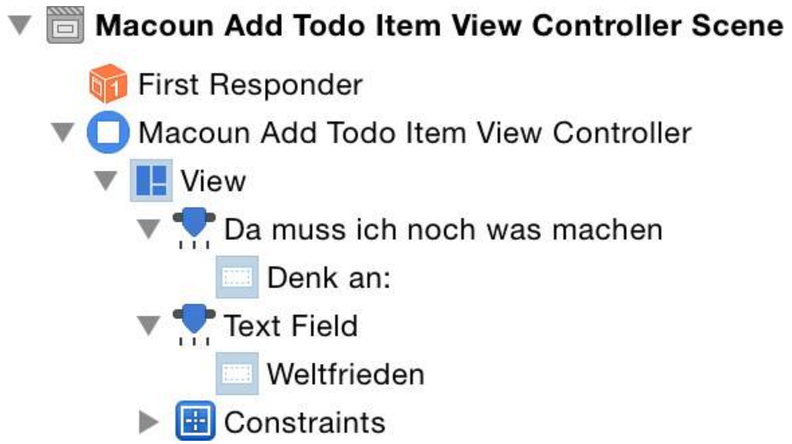
App erstellen
(jetzt wirklich)

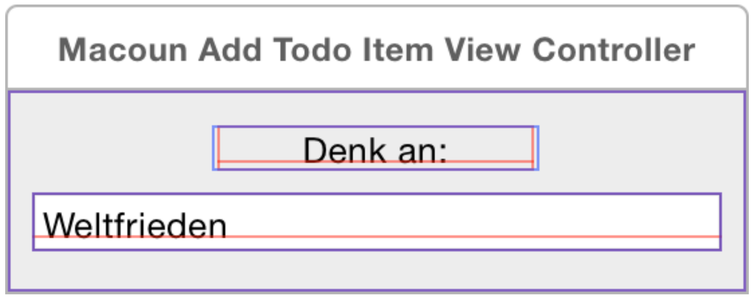



```
@interface Macoun_TodoWindowController : NSWindowController
@end
```

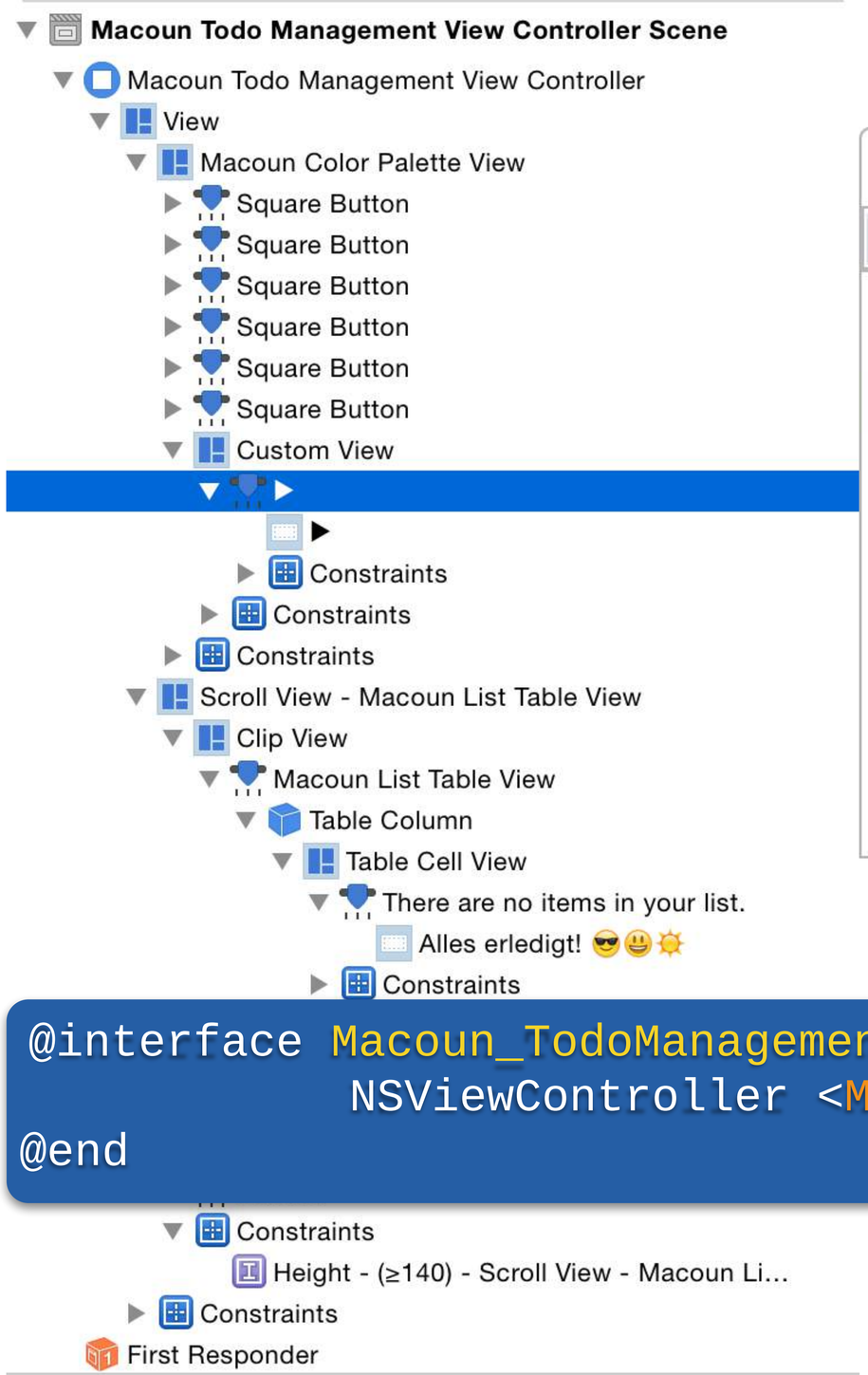


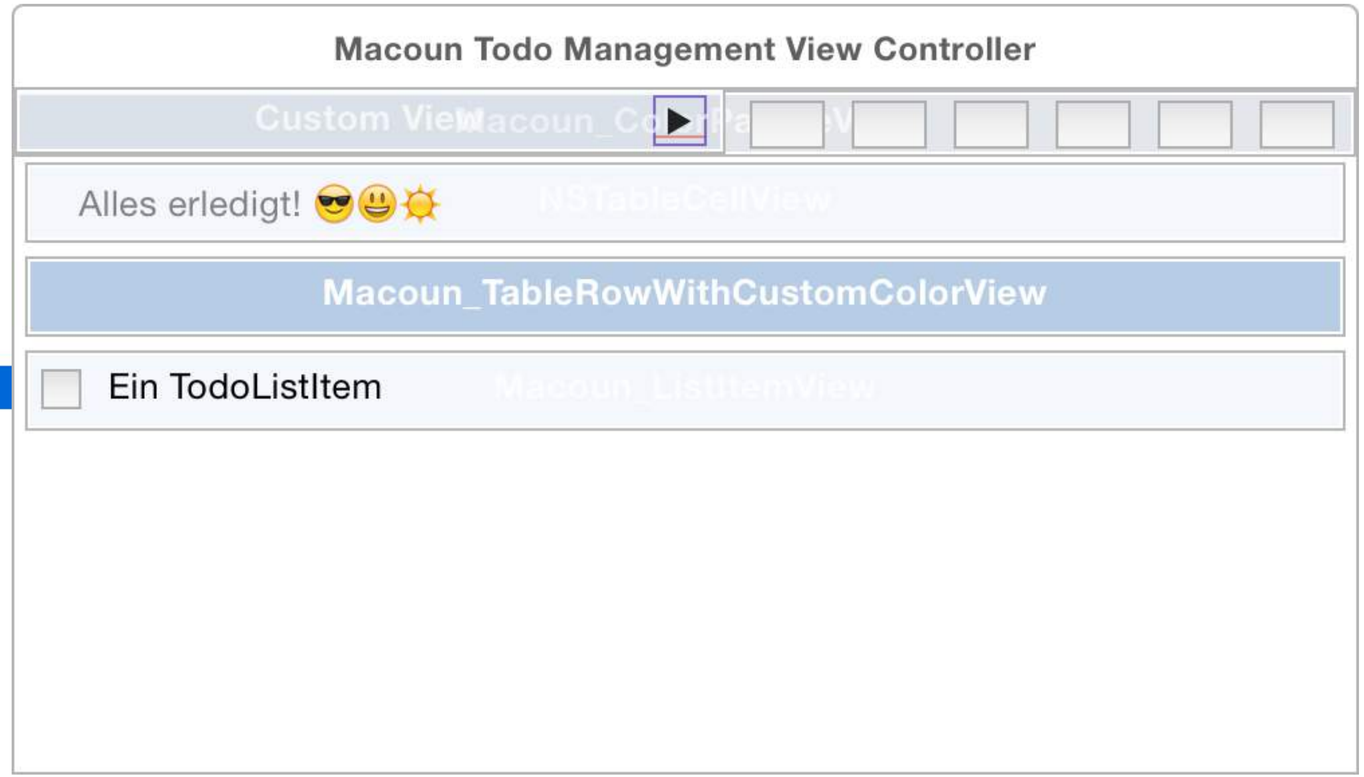

```
@interface Macoun_TodoWindowController : NSWindowController
@end
```



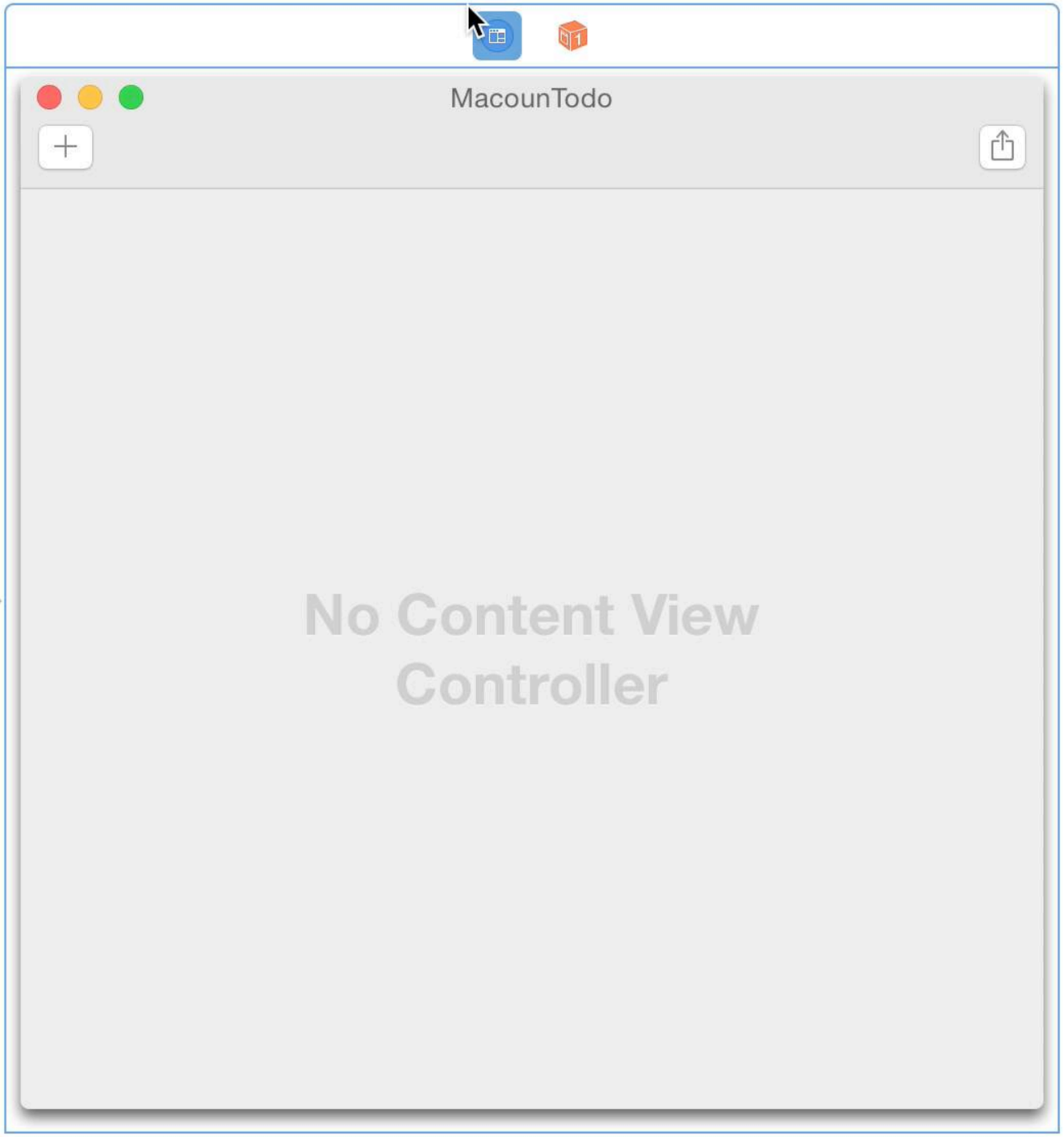


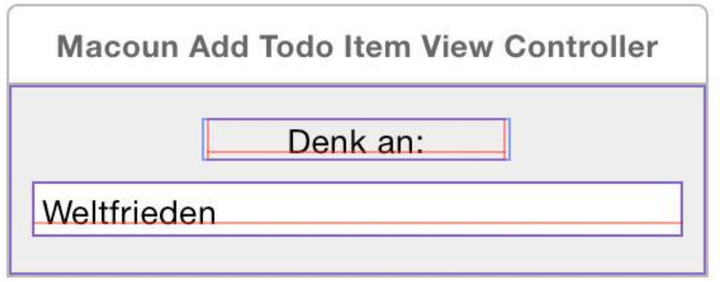

```
@protocol Macoun_AddItemViewControllerDelegate <NSObject>
- (void)addItemViewController:(Macoun_AddTodoItemViewController *)addItemVC
     didCreateNewItemWithText:(NSString *)text;
@end

@interface Macoun_AddTodoItemViewController : NSViewController
@property (weak) id<Macoun_AddItemViewControllerDelegate> delegate;
@end
```



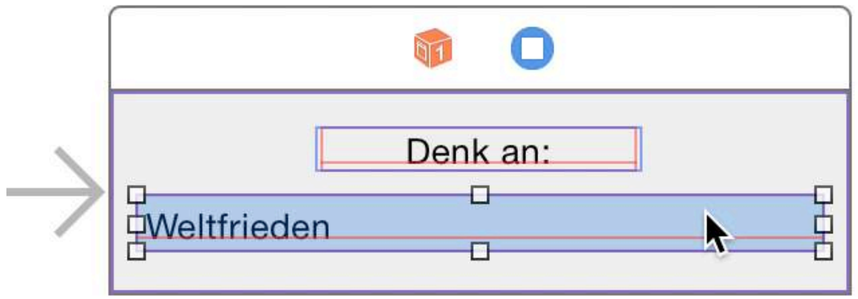



```
@interface Macoun_TodoManagementViewController :
            NSViewController <Macoun_AddItemViewControllerDelegate>
@end
```

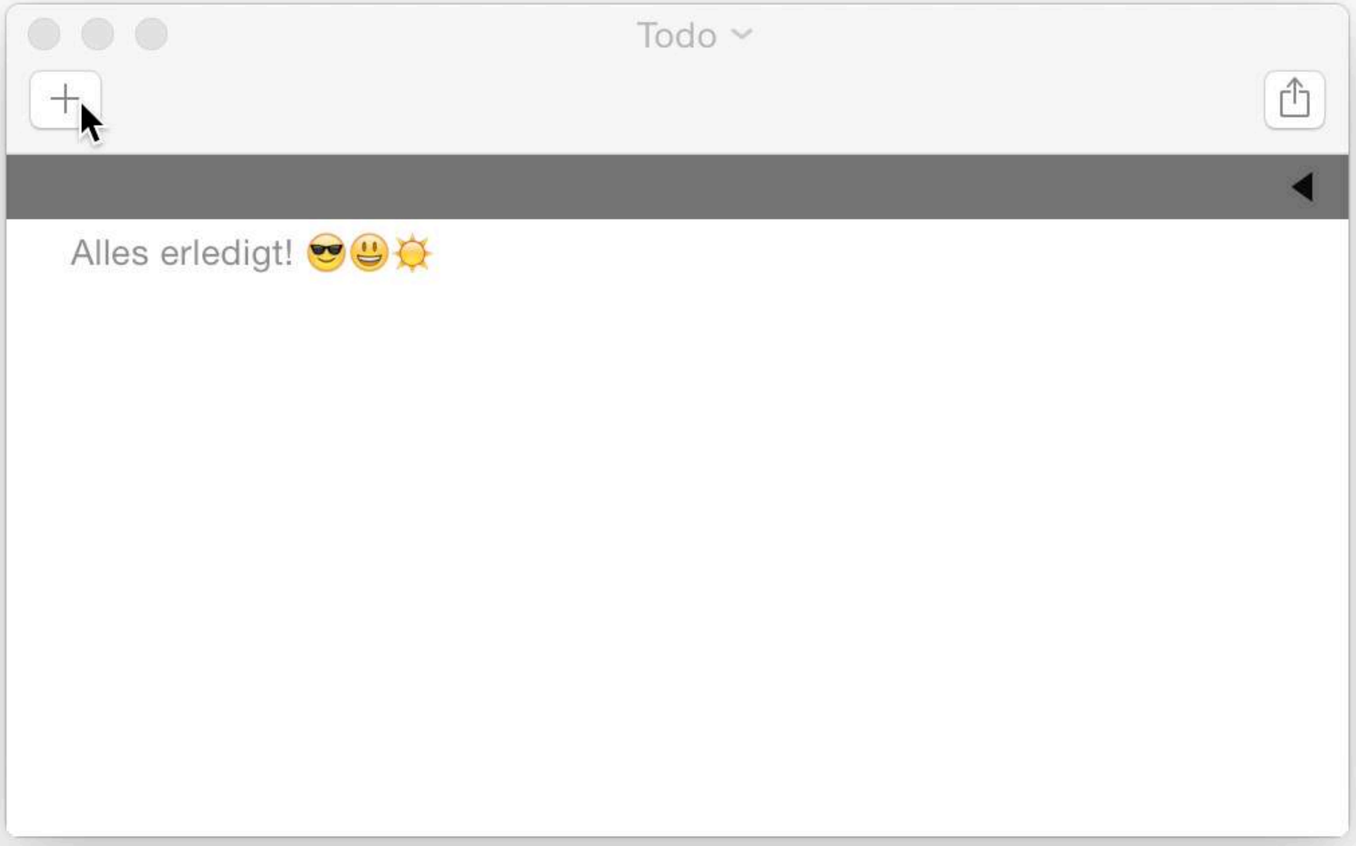

MacounTodo
No Content View Controller


Macoun Add Todo Item View Controller
Denk an:
Weltfrieden

Macoun Todo Management
Alles erledigt!
Macoun_TableRowWithCu
Ein TodoListItem

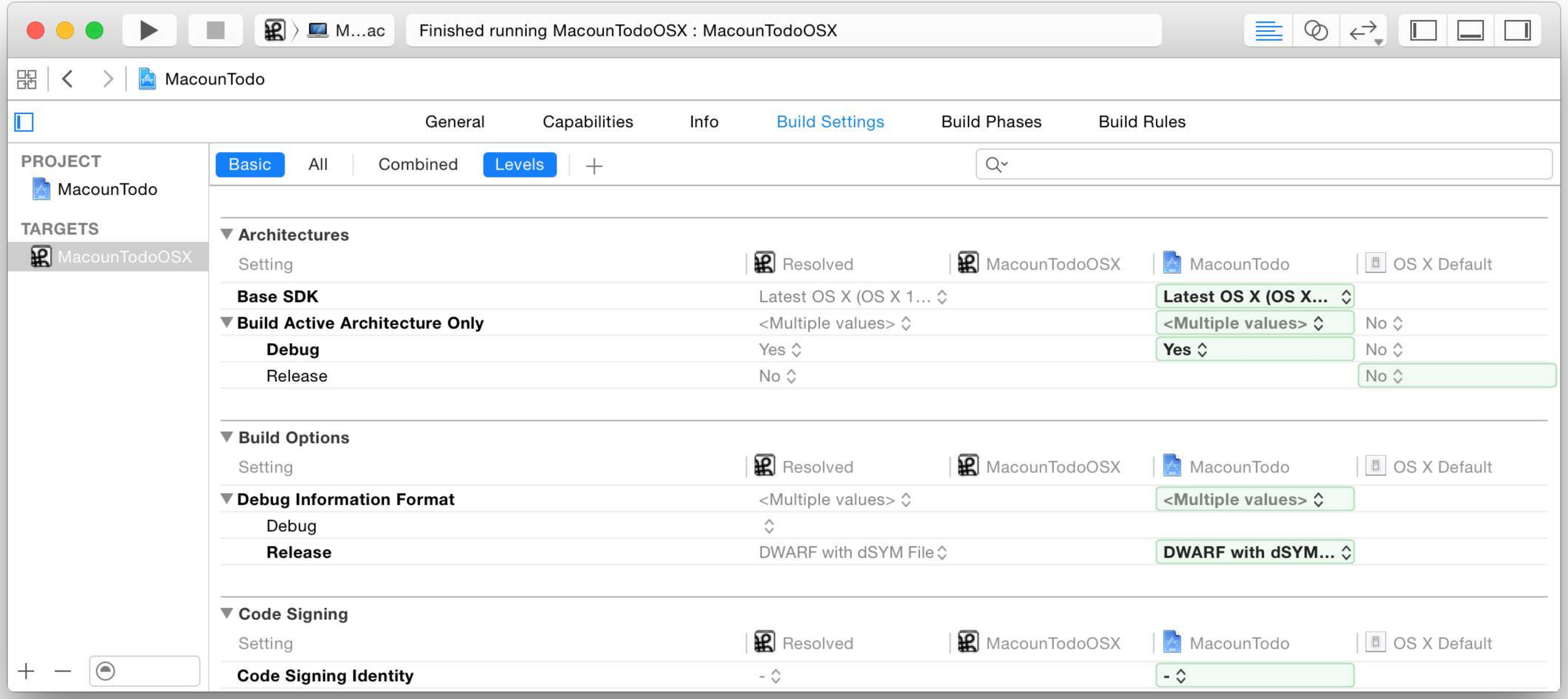


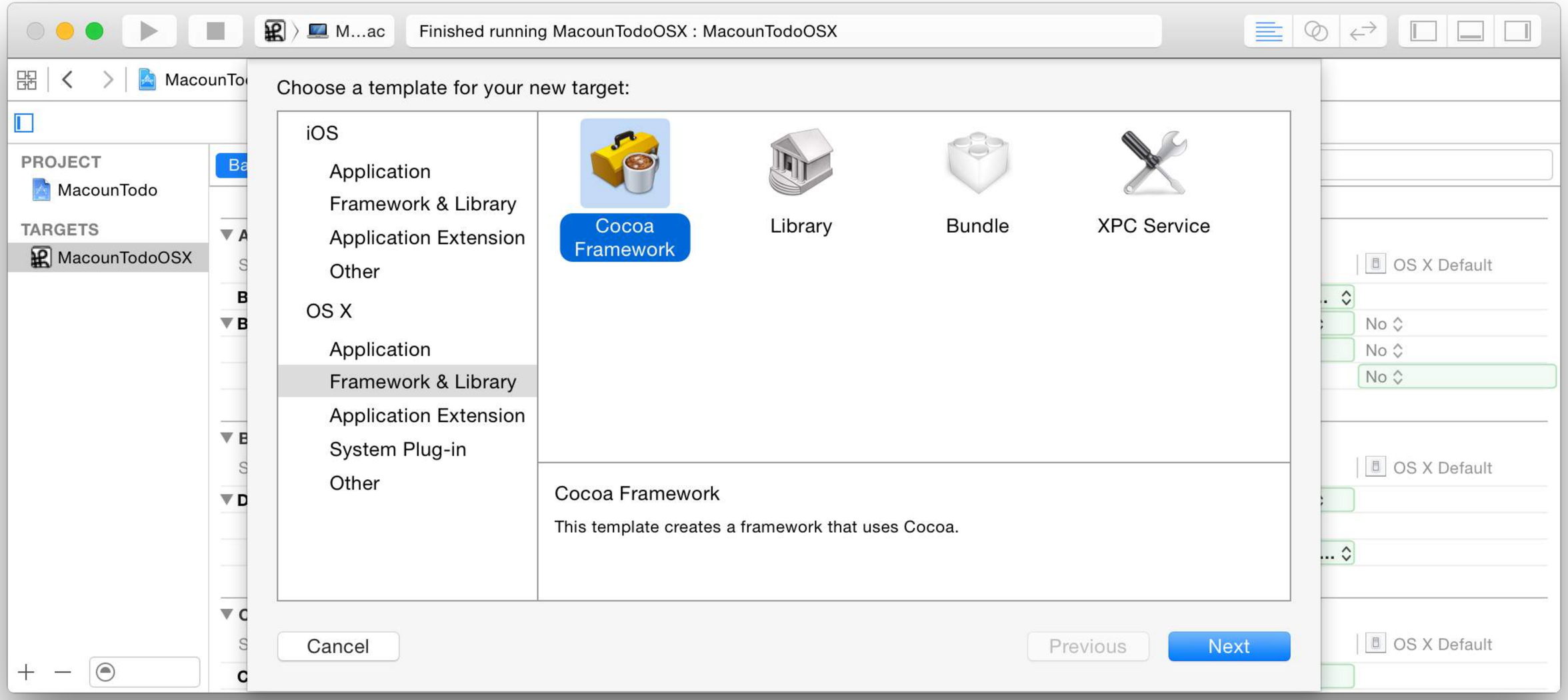

```
11  #import "Macoun_AddTodoItemViewController.h"
12  @import CoreMacounOSX;
13
14  @implementation Macoun_AddTodoItemViewController
15
16  - (IBAction)textChanged:(NSTextField *)textField {
17      NSString *cleansedString = [textField.stringValue
            stringByTrimmingCharactersInSet:[NSCharacterSet
            whitespaceAndNewlineCharacterSet]];
18
19      if (cleansedString.length > 0) {
20          [self.delegate addItemViewController:self
                didCreateNewItemWithText:cleansedString];
21      }
22
23      [self.presentingViewController dismissViewController:self];
24  }
25
26  @end
```

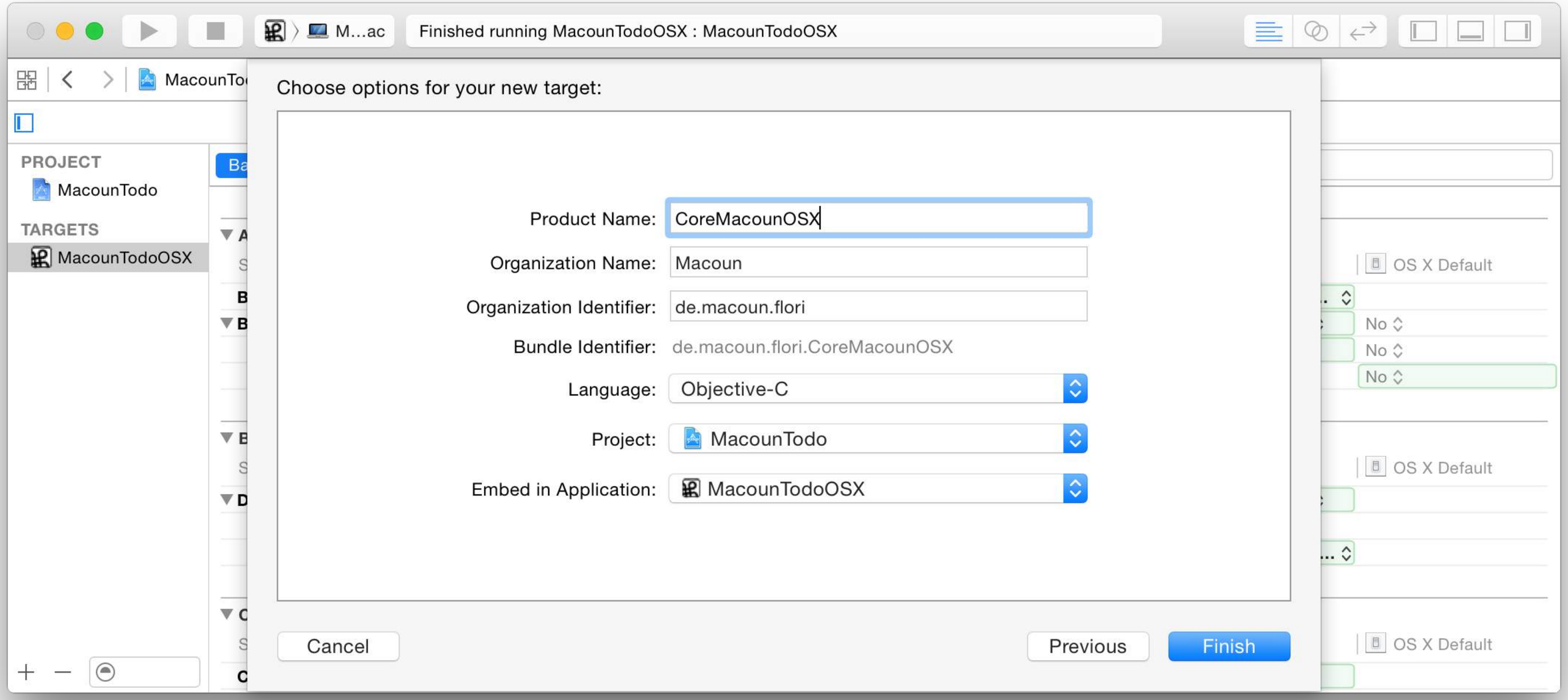

Todo
Alles erledigt! 😎😃☀️

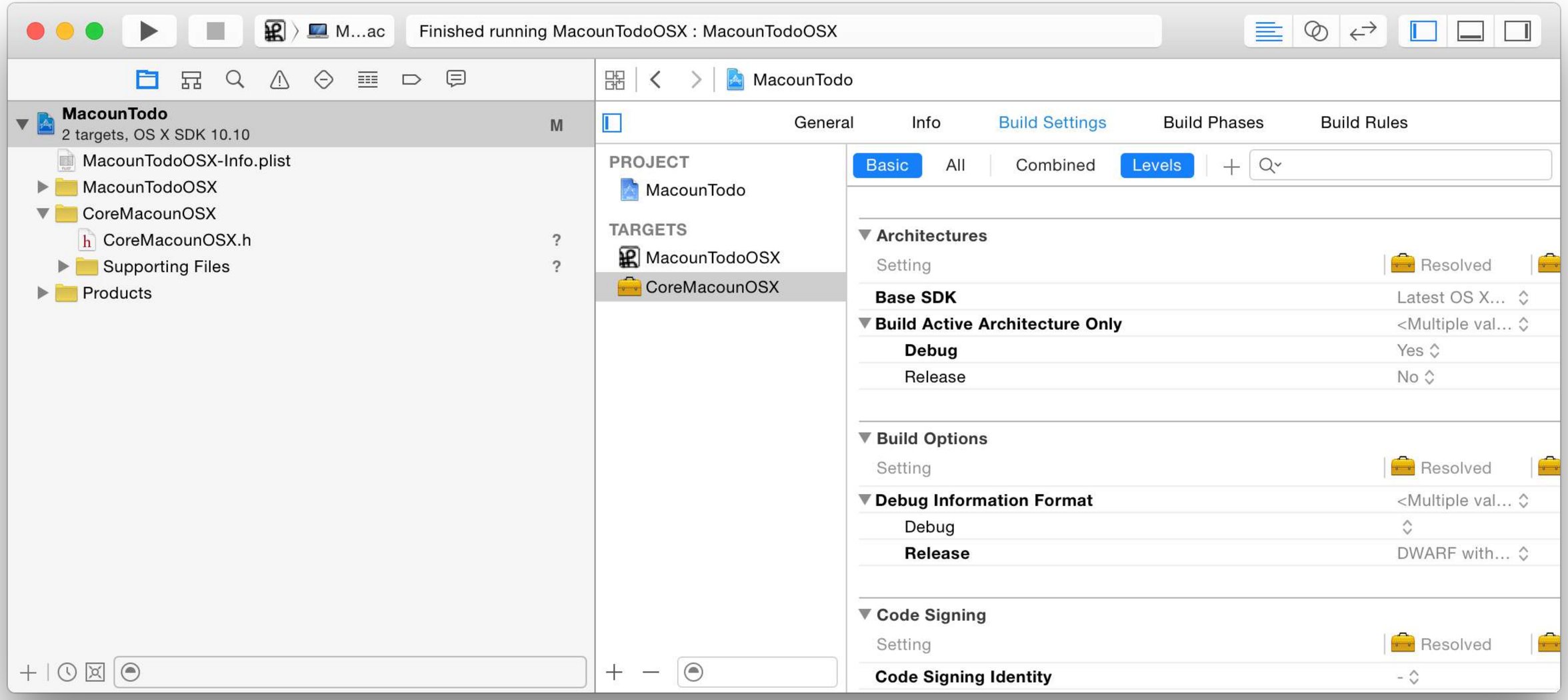
Frameworks

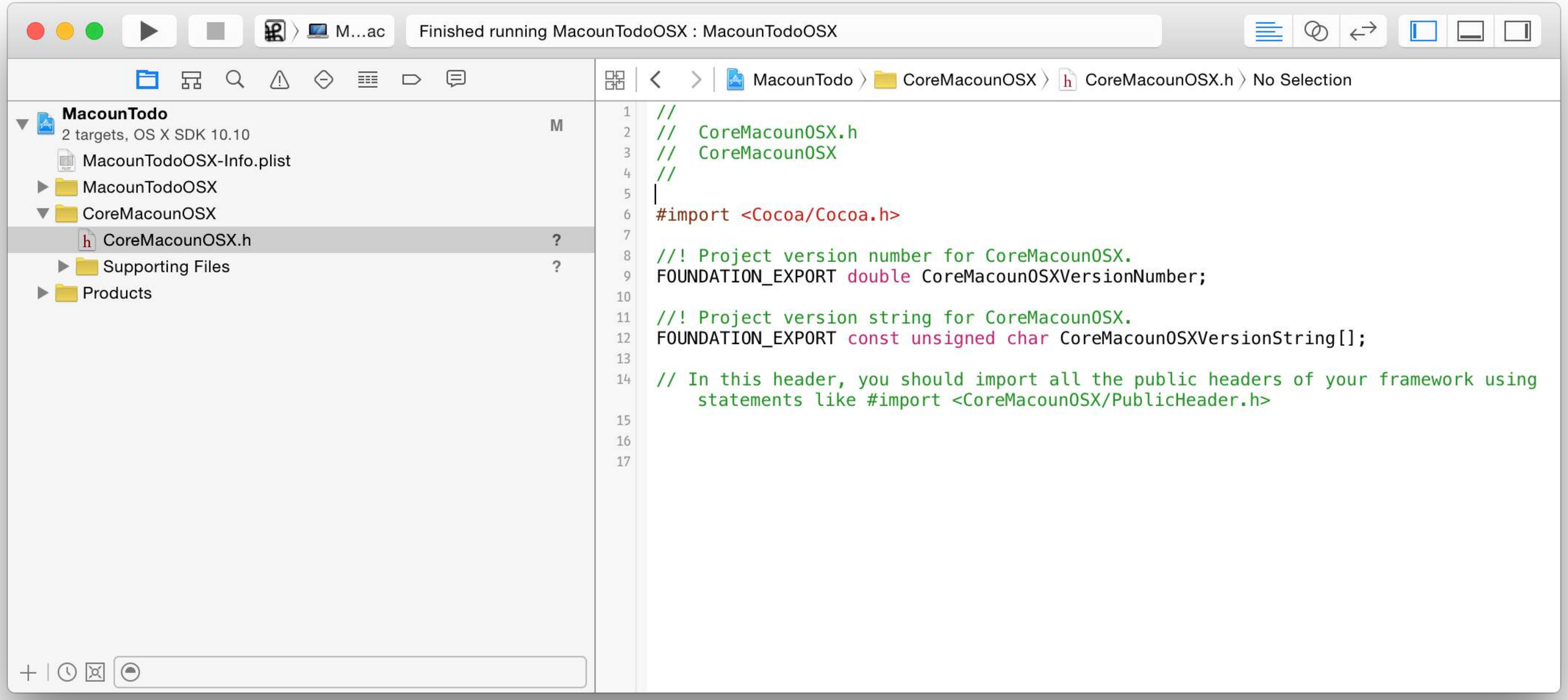
- gut für gemeinsam genutzten Code
- unter iOS können “wir” es erst seit iOS 8 erzeugen
- gibt es unter OS X schon immer
- Apple hat schon immer Frameworks für alles (mehr oder weniger)


M…ac
Finished running MacounTodoOSX : MacounTodoOSX
MacounTodo
General
Capabilities
Info
Build Settings
Build Phases
Build Rules
PROJECT
MacounTodo
TARGETS
MacounTodoOSX
Basic
All
Combined
Levels
Architectures
Setting
Resolved
MacounTodoOSX
MacounTodo
OS X Default
Base SDK
Latest OS X (OS X 1…
Latest OS X (OS X…
Build Active Architecture Only
<Multiple values>
<Multiple values>
No
Debug
Yes
Yes
No
Release
No
No
Build Options
Setting
Resolved
MacounTodoOSX
MacounTodo
OS X Default
Debug Information Format
<Multiple values>
<Multiple values>
Debug
Release
DWARF with dSYM File
DWARF with dSYM…
Code Signing
Setting
Resolved
MacounTodoOSX
MacounTodo
OS X Default
Code Signing Identity
-
-

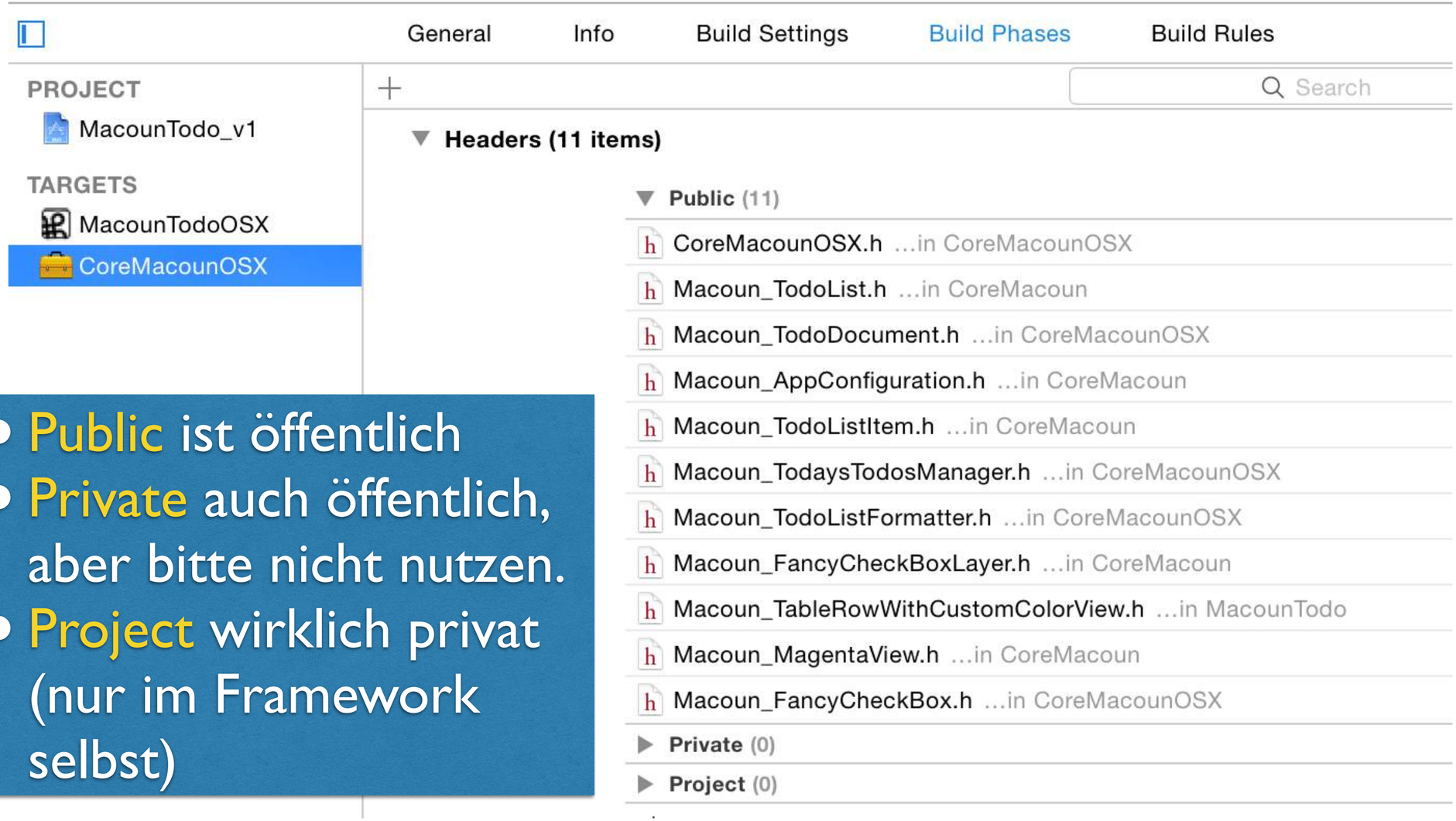

Finished running MacounTodoOSX : MacounTodoOSX
PROJECT
MacounTodo
TARGETS
MacounTodoOSX
Choose a template for your new target:
iOS
Application
Framework & Library
Application Extension
Other
OS X
Application
Framework & Library
Application Extension
System Plug-in
Other
Cocoa Framework
Library
Bundle
XPC Service
Cocoa Framework
This template creates a framework that uses Cocoa.
Cancel
Previous
Next
OS X Default
No
No
No
OS X Default
OS X Default


Finished running MacounTodoOSX : MacounTodoOSX
Choose options for your new target:
Product Name: CoreMacounOSX
Organization Name: Macoun
Organization Identifier: de.macoun.flori
Bundle Identifier: de.macoun.flori.CoreMacounOSX
Language: Objective-C
Project: MacounTodo
Embed in Application: MacounTodoOSX
Cancel
Previous
Finish
PROJECT
MacounTodo
TARGETS
MacounTodoOSX
OS X Default
No
No
No
OS X Default
OS X Default


M...ac
Finished running MacounTodoOSX : MacounTodoOSX
MacounTodo
2 targets, OS X SDK 10.10
M
MacounTodoOSX-Info.plist
MacounTodoOSX
CoreMacounOSX
CoreMacounOSX.h
Supporting Files
Products
MacounTodo
General
Info
Build Settings
Build Phases
Build Rules
PROJECT
MacounTodo
TARGETS
MacounTodoOSX
CoreMacounOSX
Basic
All
Combined
Levels
Architectures
Setting
Resolved
Base SDK
Latest OS X...
Build Active Architecture Only
<Multiple val...
Debug
Yes
Release
No
Build Options
Setting
Resolved
Debug Information Format
<Multiple val...
Debug
Release
DWARF with...
Code Signing
Setting
Resolved
Code Signing Identity
-


M...ac
Finished running MacounTodoOSX : MacounTodoOSX
MacounTodo
2 targets, OS X SDK 10.10
M
MacounTodoOSX-Info.plist
MacounTodoOSX
CoreMacounOSX
CoreMacounOSX.h
?
Supporting Files
?
Products
MacounTodo
CoreMacounOSX
CoreMacounOSX.h
No Selection
//
// CoreMacounOSX.h
// CoreMacounOSX
//
#import <Cocoa/Cocoa.h>
//! Project version number for CoreMacounOSX.
FOUNDATION_EXPORT double CoreMacounOSXVersionNumber;
//! Project version string for CoreMacounOSX.
FOUNDATION_EXPORT const unsigned char CoreMacounOSXVersionString[];
// In this header, you should import all the public headers of your framework using
statements like #import <CoreMacounOSX/PublicHeader.h>

Frameworks - Header Sichtbarkeit



- Public ist öffentlich
- Private auch öffentlich, aber bitte nicht nutzen.
- Project wirklich privat (nur im Framework selbst)

Nun zum ersten Mal WIRKLICH programmieren

Datenmodell (1/3)

public

```
@interface Macoun_TodoListItem : NSObject <NSCoding>
@property (copy) NSString *text;
@property (getter=isComplete) BOOL complete;
- (instancetype)initWithText:(NSString *)text complete:(BOOL)complete;
- (instancetype)initWithText:(NSString *)text;
- (void)refreshIdentity;
@end
```

private

```
@interface Macoun_TodoListItem()
@property NSUUID *UUID;
@end
```

Datenmodell (1/3)

```
@implementation Macoun_TodoListItem
- (instancetype)initWithText:(NSString *)text complete:(BOOL)complete {
    return [self initWithText:text complete:complete UUID:[NSUUID UUID]];
}
- (instancetype)initWithText:(NSString *)t complete:(BOOL)c UUID:(NSUUID *)UUID {
    self = [super init];
    if (self) {
        _text = [t copy];
        _complete = c;
        _UUID = UUID;
    }
    return self;
}
...
- (void)refreshIdentity {
    self.UUID = [NSUUID UUID];
}
```

Datenmodell (2/3)

```
@interface Macoun_TodoList : NSObject <NSCoding>
@property (nonatomic) Macoun_ListColor color;
@property (nonatomic, readonly) NSArray *allItems;
@property (readonly) NSInteger count;
@property (readonly, getter=isEmpty) BOOL empty;
- (instancetype)initWithColor:(Macoun_ListColor)color items:(NSArray *)items;
- (instancetype)init; //einfach Grau und eine leere Liste
- (NSIndexSet *)insertItems:(NSArray *)itemsToInsert;
- (void)insertItem:(Macoun_TodoListItem *)item atIndex:(NSInteger)index;
- (NSInteger)insertItem:(Macoun_TodoListItem *)item;
- (Macoun_TodoListItem *)objectAtIndexedSubscript:(NSUInteger)index;
- (NSArray *)objectForKeyedSubscript:(NSIndexSet *)indexes;
- (NSInteger)indexOfItem:(Macoun_TodoListItem *)item;
...
```

Datenmodell (3/3)

```
@protocol Macoun_TodoDocumentDelegate <NSObject>
- (void)listDocumentDidChangeContents:(Macoun_TodoDocument *)document;
@end


@interface Macoun_TodoDocument : NSDocument
@property (weak) id<Macoun_TodoDocumentDelegate> delegate;
@property Macoun_TodoList *list;
@end
```

Datenmodell (3/3)

```
@protocol Macoun_TodoDocumentDelegate <NSObject>
- (void)listDocumentDidChangeContents:(Macoun_TodoDocument *)document;
@end


@interface Macoun_TodoDocument : NSDocument
@property (weak) id<Macoun_TodoDocumentDelegate> delegate;
@property Macoun_TodoList *list;
@end
```

NSDocument - Init/Config (1/4)

```
@implementation Macoun_TodoDocument
- (instancetype)init {
    self = [super init];
    if (self) {
        _list = [Macoun_TodoList new];
    }
    return self;
}

+ (BOOL)autosavesInPlace {
    return YES;
}

- (NSString *)defaultDraftName {
    return @"Todo";
}
```

NSDocument - GUI (2/4)

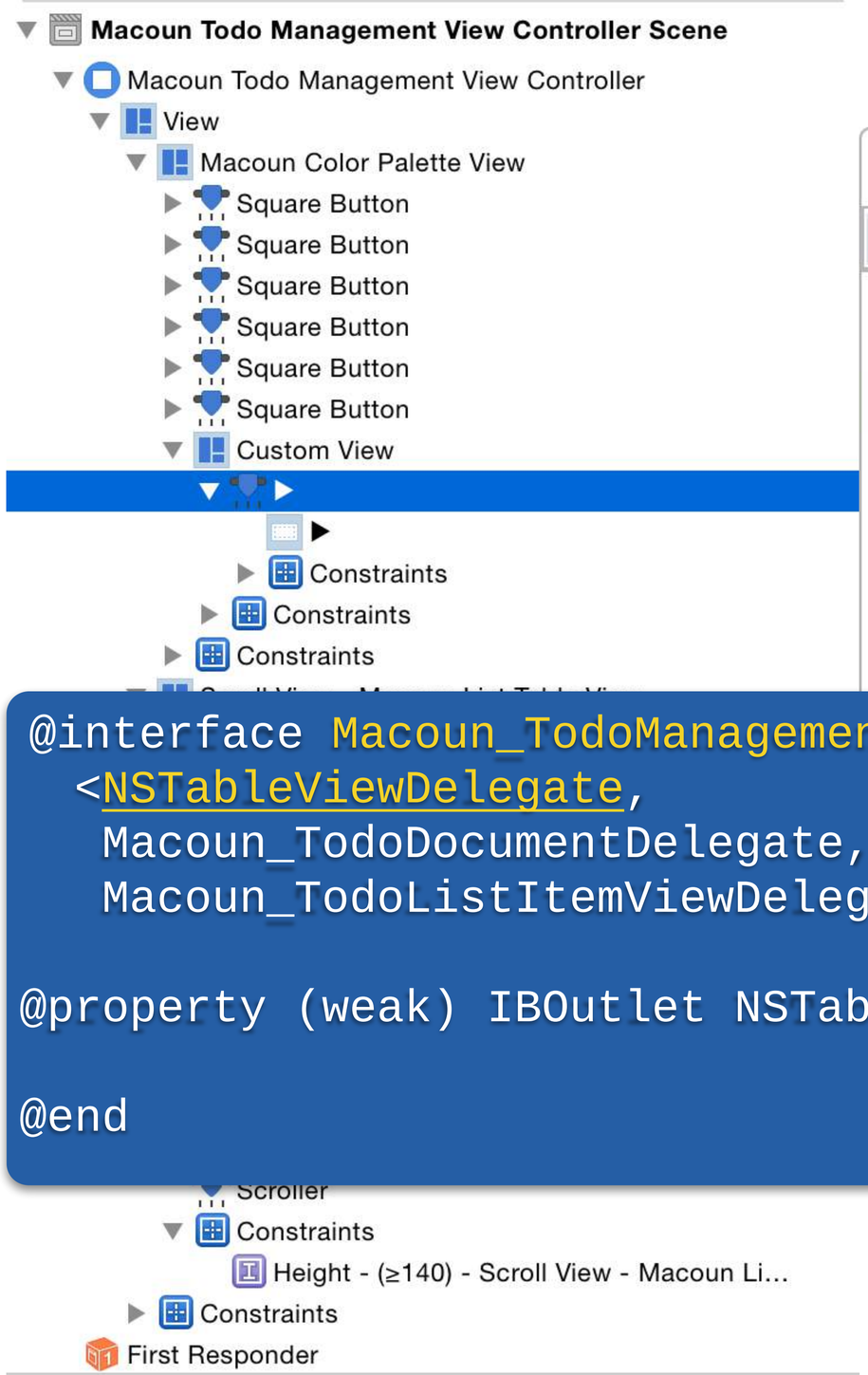
```
- (void)makeWindowControllers {
    [super makeWindowControllers];

    NSStoryboard *sb = [NSStoryboard storyboardWithName:@"Storyboard" bundle:nil];

    NSWindowController *windowController = [sb instantiateInitialController];

    [self addWindowController:windowController];
}
```

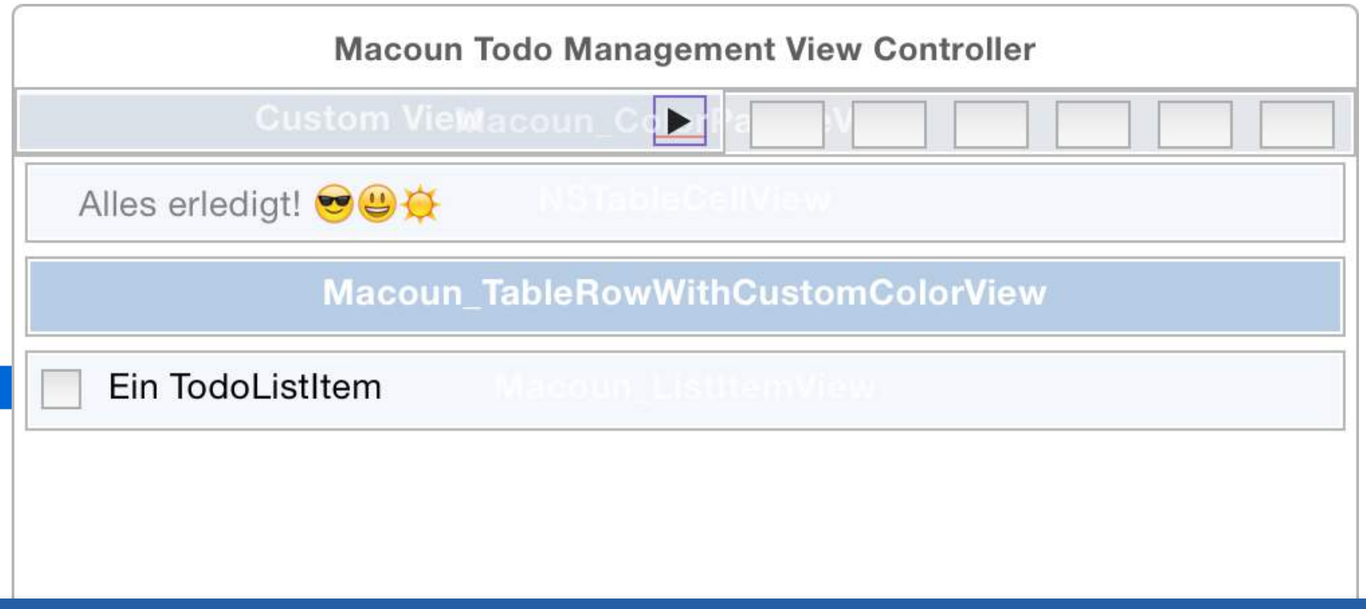
NSDocument - Laden (3/4)

```
- (BOOL)readFromData:(NSData *)data ofType:(NSString *)typeName
              error:(NSError *__autoreleasing *)outError {
    Macoun_TodoList *deserializedL = [NSKeyedUnarchiver unarchiveObjectWithData:data];

    if (deserializedL) {
        *outError = nil;
        self.list = deserializedL;
        [self.delegate listDocumentDidChangeContents:self];
        return YES;
    }

    if (outError) {
        *outError = [NSError errorWithDomain:NSCocoaErrorDomain ...];
    }

    return NO;
}
```

NSDocument - Speichern (4/4)

```
- (NSData *)dataOfType:(NSString *)typeName
                 error:(NSError *__autoreleasing *)outError {
    return [NSKeyedArchiver archivedDataWithRootObject:self.list];
}
```

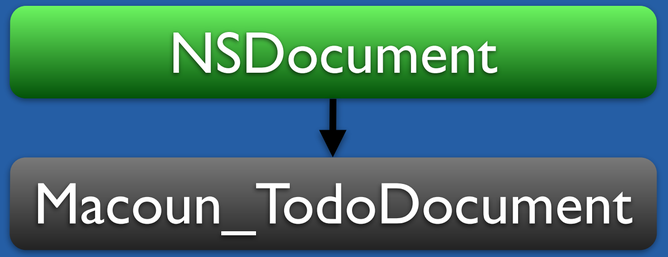





```
@interface Macoun_TodoManagementViewController()
  <NSTableViewDelegate,
   Macoun_TodoDocumentDelegate,
   Macoun_TodoListItemViewDelegate, ...>

@property (weak) IBOutlet NSTableView *tableView;

@end
```

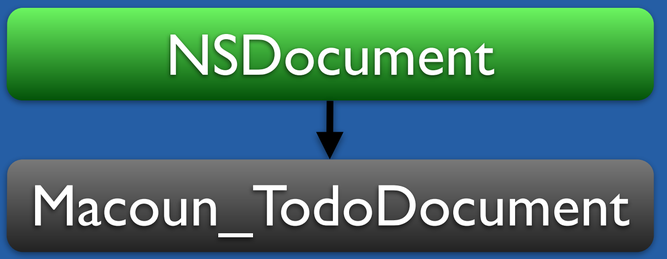
Macoun_TodoManagementViewController (1 von vielen ❤️)

```
- (void)viewDidAppear {
    [super viewDidAppear];
    self.document.delegate = self;
}

- (void)setDocument:(Macoun_TodoDocument *)document {
    _document = document;
    document.delegate = self;
    [self reloadListUI];
}
```



Macoun_TodoManagementViewController (2 von vielen ❤️)

```
- (NSUndoManager *)undoManager {
    return [self.document undoManager];
}
```



Macoun_TodoManagementViewController (3 von vielen ❤️)

```
- (NSView *)tableView:(NSTableView *)table
   viewForTableColumn:(NSTableColumn *)tableColumn row:(NSInteger)row {

    if (self.document.list.isEmpty) {
        return [table makeViewWithIdentifier:Macoun_NoTodoItemID owner:nil];
    }

    Macoun_TodoListItem *item = self.document.list[row];

    Macoun_ListItemView *view = [table makeViewWithIdentifier:Macoun_TodoItemID owner:nil
    view.completed = item.isComplete;
    view.tintColor = Macoun_ColorMagicFromListColor(self.document.list.color);
    view.stringValue = item.text;
    view.delegate = self;
    return view;
}
```

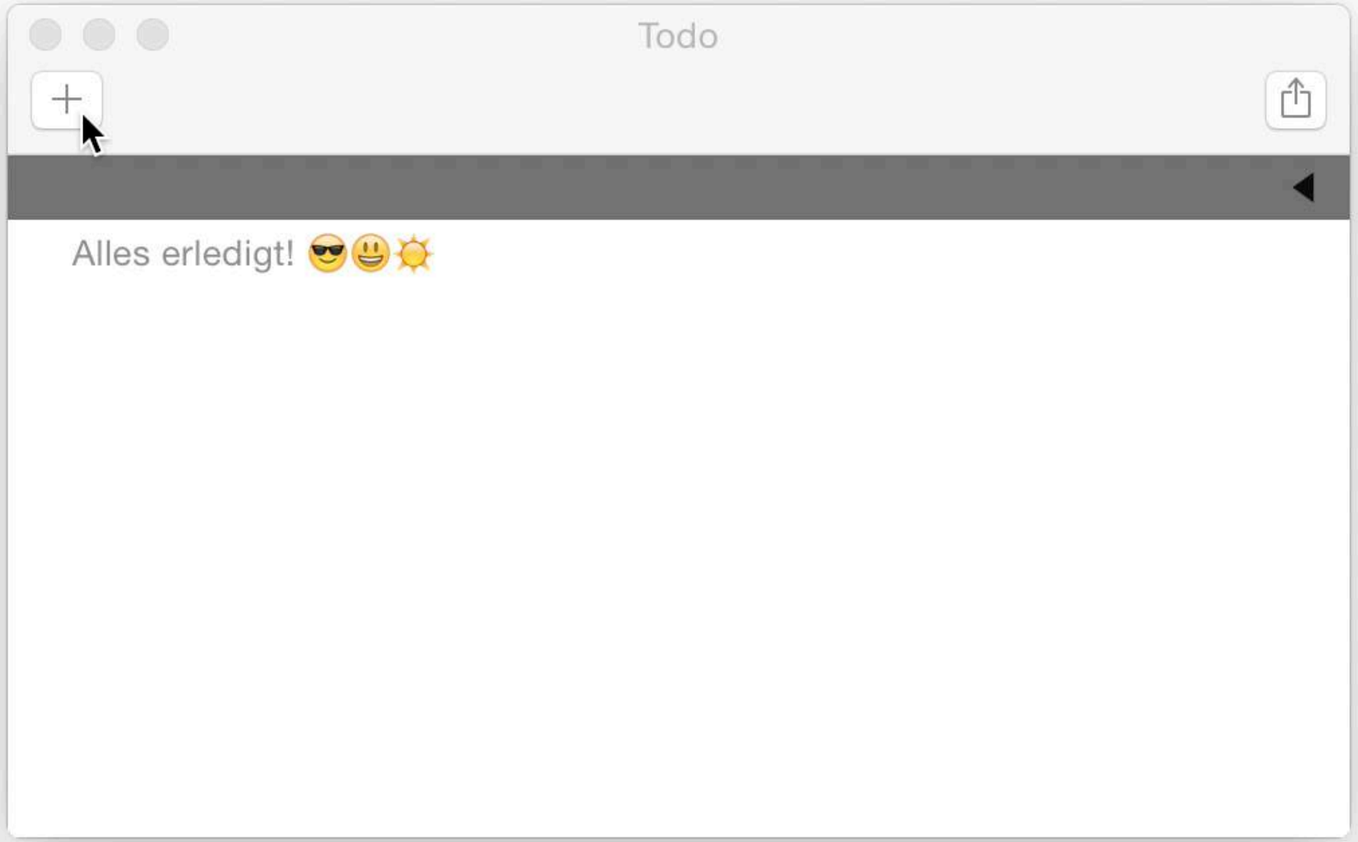
Macoun_TodoManagementViewController (4 von vielen ❤️)

```
@protocol Macoun_AddItemViewControllerDelegate <NSObject>
- (void)addItemViewController:(Macoun_AddTodoItemViewController *)addItemVC
     didCreateNewItemWithText:(NSString *)text;
@end
```

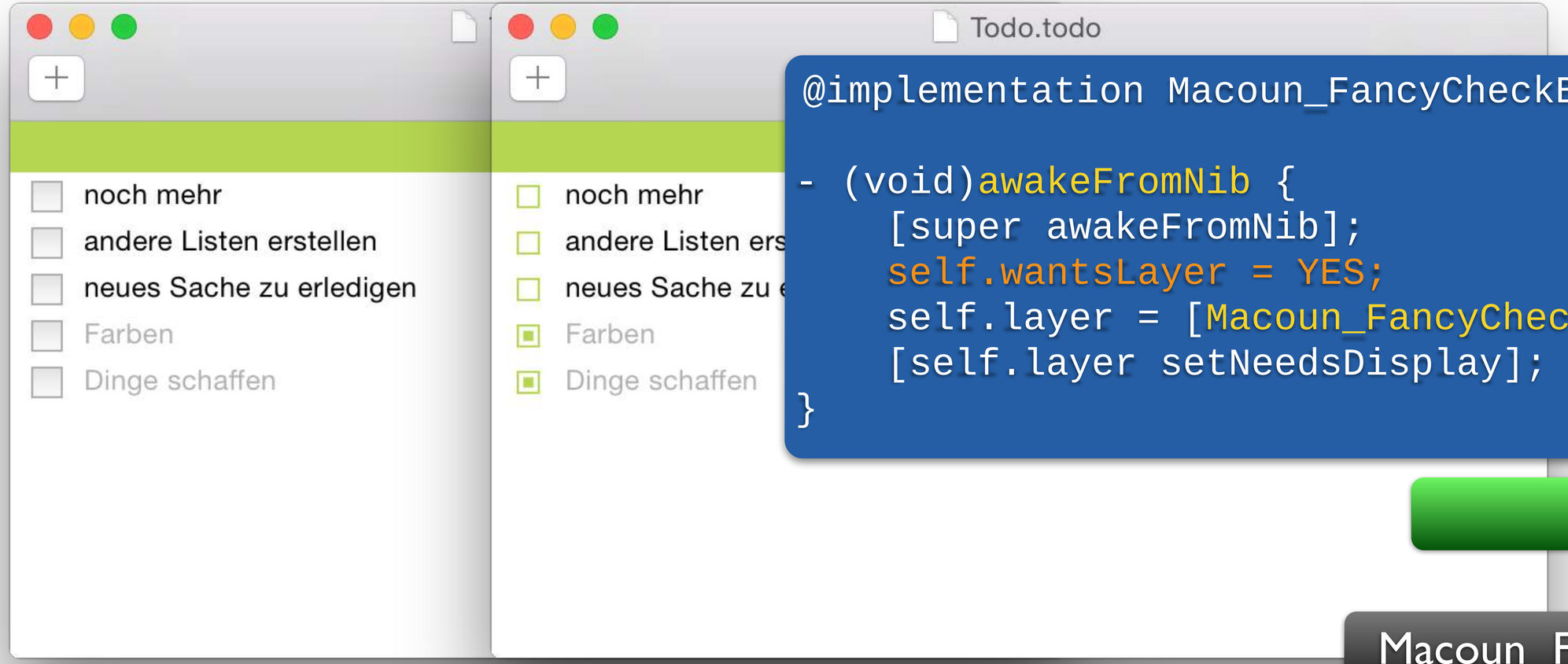
Macoun_AddTodoItemViewController

```
- (void)addItemViewController:(Macoun_AddTodoItemViewController *)addItemVC
     didCreateNewItemWithText:(NSString *)text
{
    Macoun_TodoListItem *item = [[Macoun_TodoListItem alloc] initWithText:text];

    [self insertItems:@[item] withPreferredIndexes:nil];
}
```

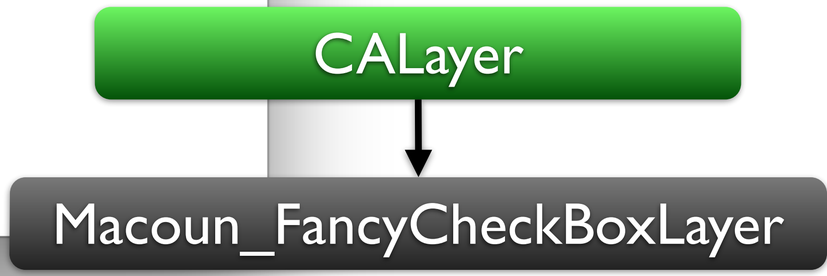
Macoun_TodoManagementViewController


Todo
Alles erledigt! 😎😃☀️

Bonus - CheckBox + Layer




```
@implementation Macoun_FancyCheckBox

- (void)awakeFromNib {
    [super awakeFromNib];
    self.wantsLayer = YES;
    self.layer = [Macoun_FancyCheckBoxLayer layer];
    [self.layer setNeedsDisplay];
}
```



Bonus - Complete + Animation (1/2)

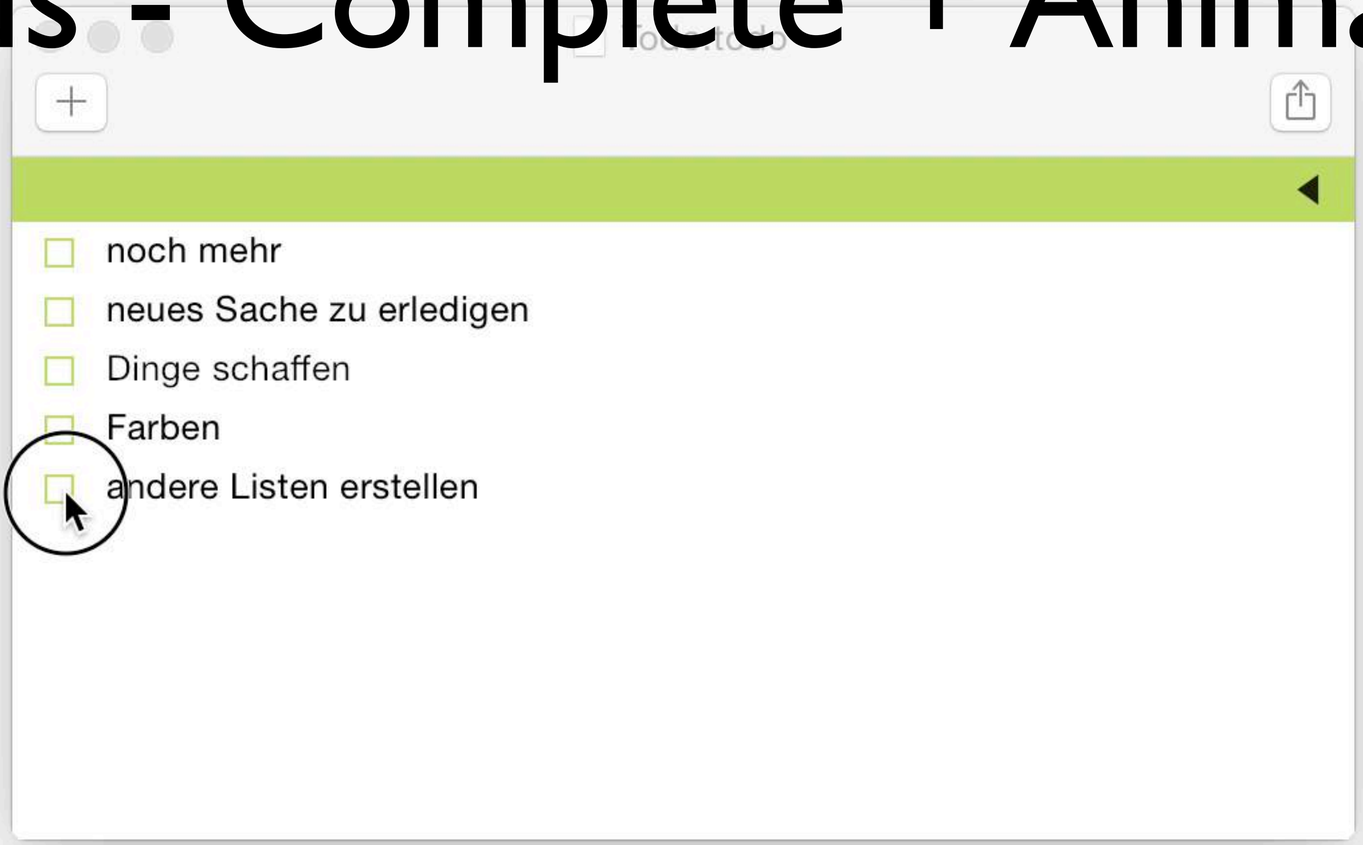
```
@protocol Macoun_TodoListItemViewDelegate <NSObject>
- (void)listItemViewDidToggleCompletionState:(Macoun_ListItemView *)listItemView;
@end

@interface Macoun_ListItemView : NSTableCellView
@property (weak) id<Macoun_TodoListItemViewDelegate> delegate;
@property (getter=isComplete) BOOL completed;
@property NSColor *tintColor;
@property (copy) NSString *stringValue;
@end
```

```
- (IBAction)statusCheckBoxButtonClicked:(NSButton *)sender {
    [self.delegate listItemViewDidToggleCompletionState:self];
}
```

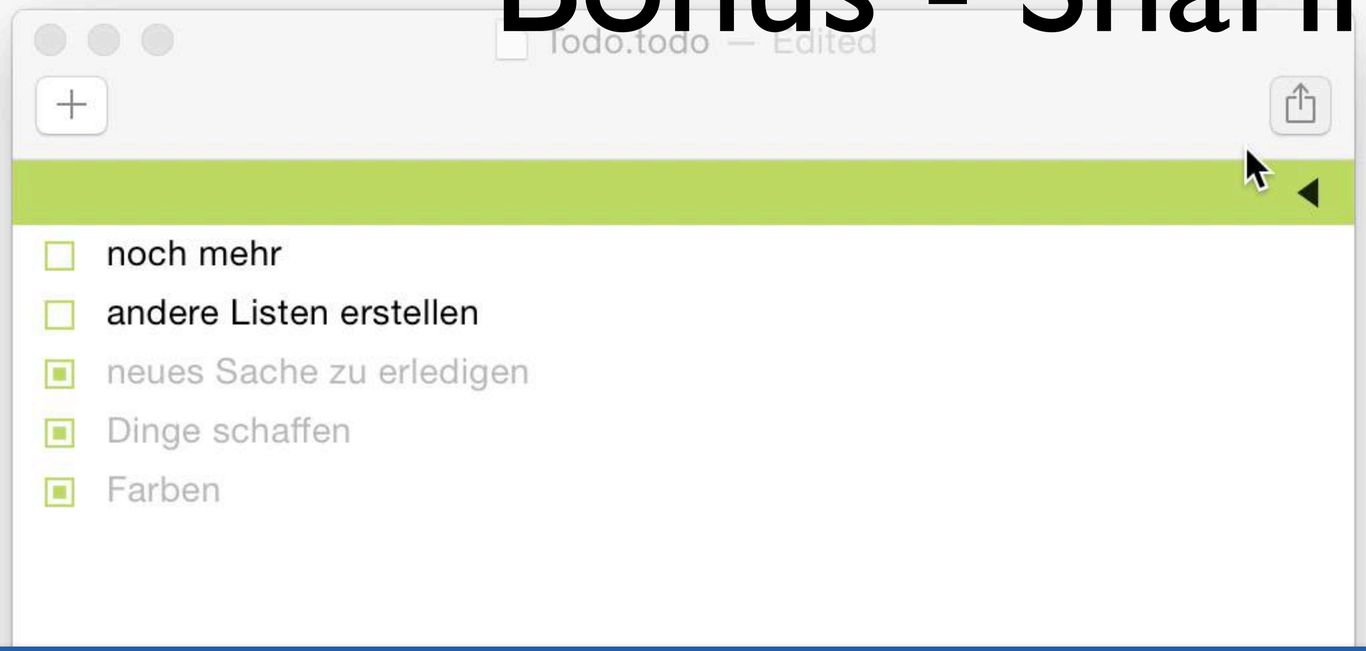
Bonus - Complete + Animation (2/2)

```
- (void)listItemViewDidToggleCompletionState:(Macoun_ListItemView *)listItemView {
    NSInteger row = [self.tableView rowForView:listItemView];
    Macoun_TodoListItem* item = self.document.list[row]

    [self.tableView beginUpdates];
    NSInteger currentIndex = [self.document.list indexOfItem:item];
    NSInteger toIndex      = [self.document.list toggleMagicItem:item
                                    withPreferredDestinationIndex:NSNotFound];

    [self.tableView moveRowAtIndex: currentIndex toIndex: toIndex];
    listItemView.completed = item.isComplete;
    [self.tableView endUpdates];

    ...
}
```

Bonus - Complete + Animation



Bonus - Sharing



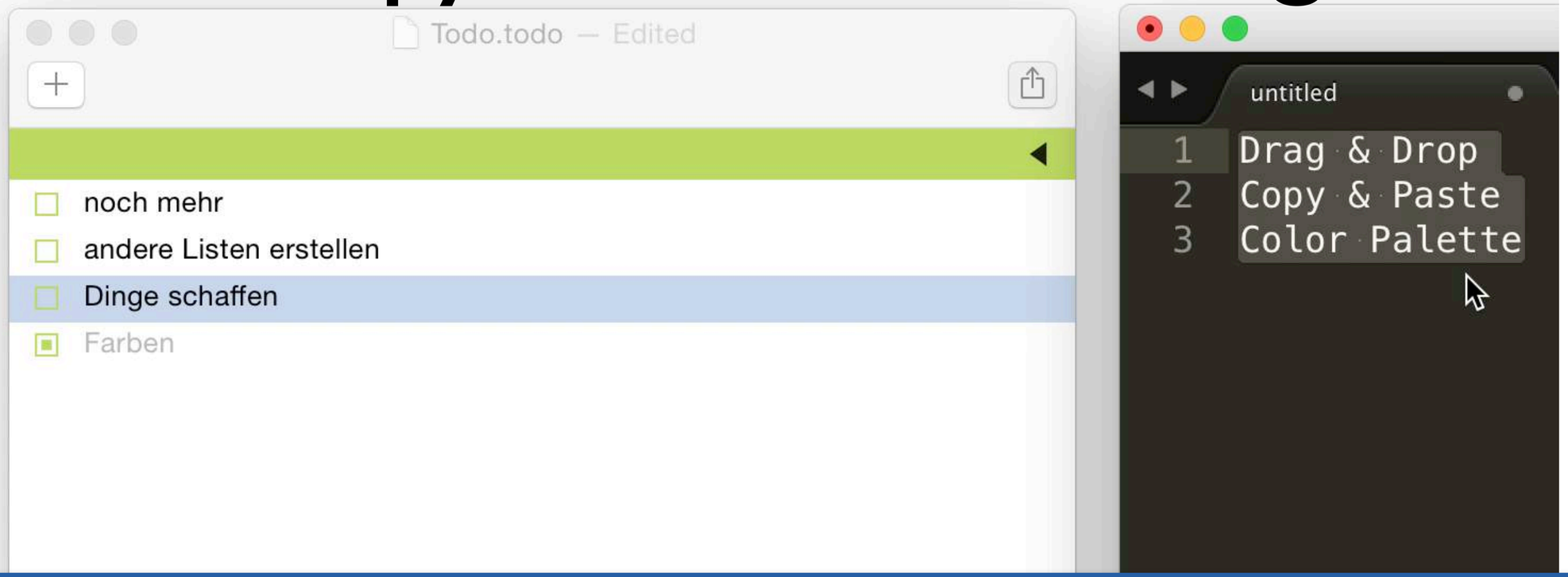

```
- (IBAction)shareDocument:(NSButton *)sender {
    Macoun_TodoDocument *document = self.document;
    NSString *s = [document asStringMagicMethod];
    NSSharingServicePicker *share = [[NSSharingServicePicker alloc] initWithItems:@[s]];
    [share showRelativeToRect:NSZeroRect ofView:sender preferredEdge:NSMinYEdge];
}
```

Bonus - Copy & Paste

```
- (void)copy:(id)sender {
    NSIndexSet *selectedRowIndexes = self.tableView.selectedRowIndexes;

    if (selectedRowIndexes.count > 0) {
        NSArray *items = self.document.list[selectedRowIndexes];
        NSString* itemsString = [items asStringMagicMethod];

        NSPasteboard* pasteboard = [NSPasteboard generalPasteboard]
        [pasteboard setString:itemsString forType:NSPasteboardTypeString];
    }
}

- (void)paste:(id)sender { ... }
- (void)cut:(id)sender { ... }
```

UIResponderStandardEditActions

Bonus - Copy & Paste + Drag n Drop




```
- (NSDragOperation)tableView:(NSTableView *)tableView
                validateDrop:(id<NSDraggingInfo>)info
                 proposedRow:(NSInteger)row
       proposedDropOperation:(NSTableViewDropOperation)dropOperation {}
```

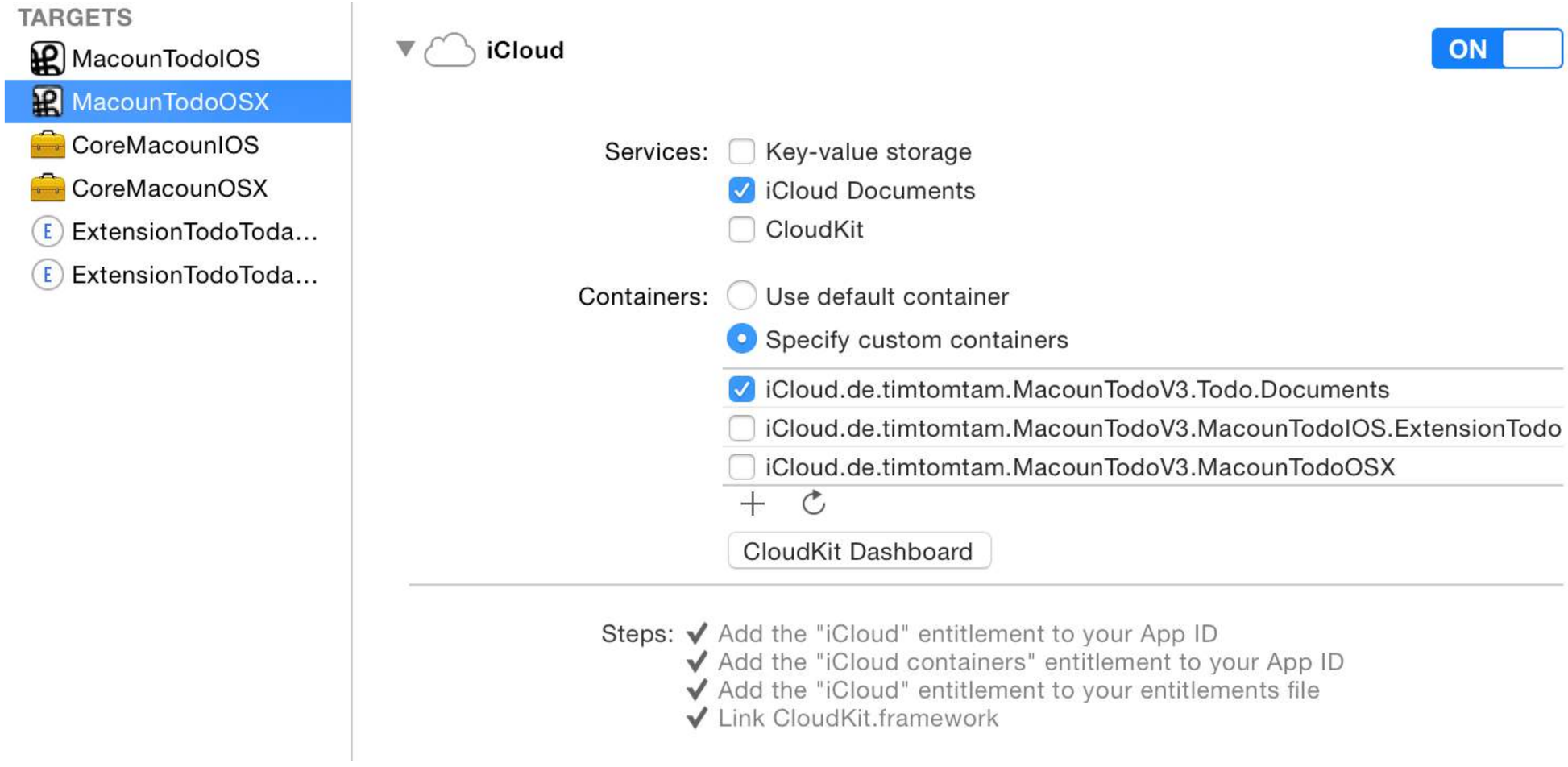
Bonus - ColorPaletteView

```
 @interface Macoun_ColorPaletteView()
@property (weak) IBOutlet NSLayoutConstraint *overlayLayoutConstraint;
@end

- (IBAction)colorButtonClicked:(NSButton *)sender {
    CGFloat overLayWidth = (self.expanded) ? 0.0f : 235.0f;
    [NSAnimationContext runAnimationGroup:^(NSAnimationContext *context) {
        // Customize the animation parameters.
        context.duration = 0.25;
        context.timingFunction = [CAMediaTimingFunction
functionWithName:kCAMediaTimingFunctionEaseInEaseOut];

        self.overlayLayoutConstraint.animator.constant = overLayWidth;
        self.selectedColor = newSelectedColor;
    } completionHandler:^{
        if (colorsAreDifferent) {
            [self.delegate colorPaletteViewDidChangeSelectedColor:self];
        }
    }];
```

Bonus - ColorPaletteView

NSDocument

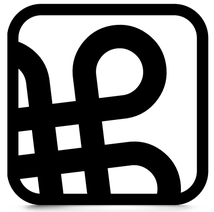
UIDocument


TARGETS
MacounTodoIOS
MacounTodoOSX
CoreMacounIOS
CoreMacounOSX
ExtensionTodoToda…
ExtensionTodoToda…
iCloud
ON
Services:
Key-value storage
iCloud Documents
CloudKit
Containers:
Use default container
Specify custom containers
iCloud.de.timtomtam.MacounTodoV3.Todo.Documents
iCloud.de.timtomtam.MacounTodoV3.MacounTodoIOS.ExtensionTodo
iCloud.de.timtomtam.MacounTodoV3.MacounTodoOSX
CloudKit Dashboard
Steps:
Add the "iCloud" entitlement to your App ID
Add the "iCloud containers" entitlement to your App ID
Add the "iCloud" entitlement to your entitlements file
Link CloudKit.framework

Carrier
9:49 AM
Extras
MacounTODO

Fragen?

Vielen Dank



Macoun