

iPhone アプリ開発のコツとツボ 35

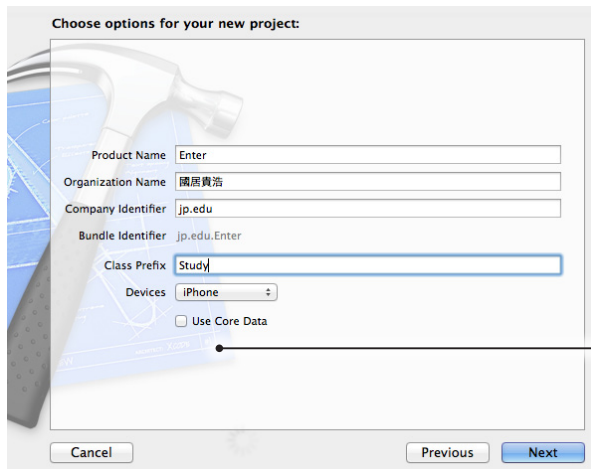
Xcode 5 および iOS 7 対応

2013.10.28

ARC について

ダウンロードしたサンプルプロジェクト自体は、Xcode 5 でも Run できるようになっています。

ですが、Xcode 5 からはプロジェクトを作る際に、ARC の使用 / 非使用を選べなくなりました。いつでも ARC が使用状態となります。



Use Automatic Reference Countingの項目が無くなった

そのためプロジェクトを本に従って、自分でアプリケーションを作っていこうと思われる方は、次の2つの方法のどちらかで対応してください。

- 1、自分でサンプルソースを ARC 対応の記述に変換して書き込んでいく。
- 2、プロジェクトの設定を手動で変更し ARC を使用しないようにする。

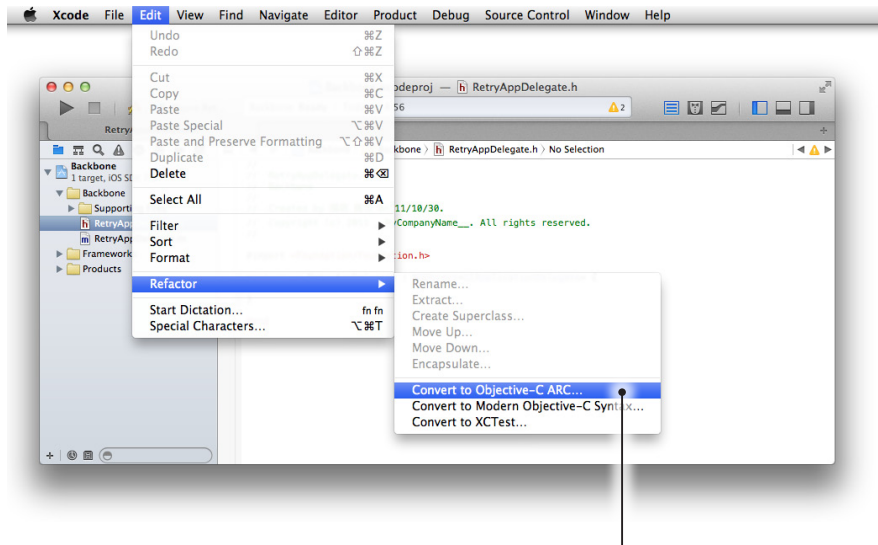
1、自分でサンプルソースを ARC 対応の記述に変換して書き込んでいく

1 の場合は retain、release、autorelease メッセージを送っているところを記述できない点と、-dealloc メソッドでは super 側を呼び出せないという点に気をつけて書き込んでいきます。

ARC 使用 / 非使用のソースの違いは、ダウンロードしたサンプルプロジェクトを Xcode 5 で開き、Edit → Refactor → Convert to Objective-C ARC…メニューを選ぶ事でも確認できます。

例えばサンプルプロジェクト 01/Backbone の Rotate.xcodeproj を例のあげると、次のような作業です。

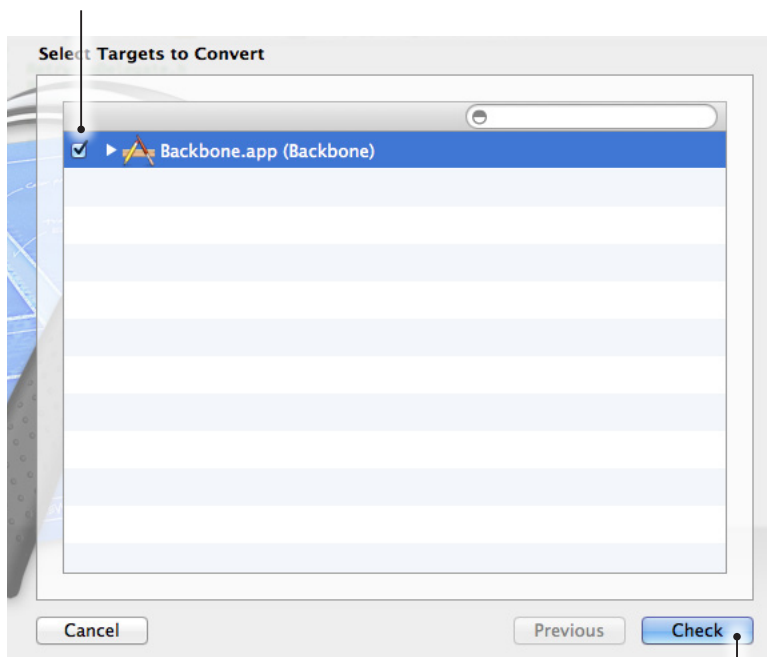
最初に Rotate.xcodeproj を開き、ワークスペースウィンドウが表示された状態で Edit → Refactor → Convert to Objective-C ARC…メニューを選びます。



Edit → Refactor → Convert to Objective-C ARC…メニューを選ぶ

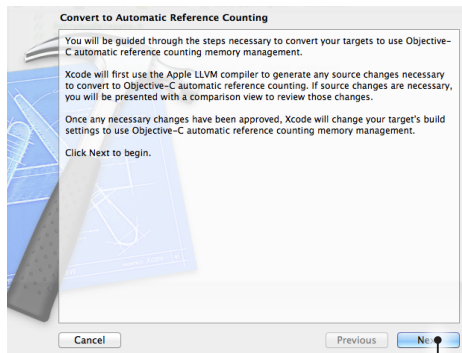
メニューを選ぶと、どのソースを ARC 対応にコンバートするか聞かれるので、アプリケーション項目横のチェックを On にしてから、Check ボタンをクリックします。

1 アプリケーション横のチェックを On にする



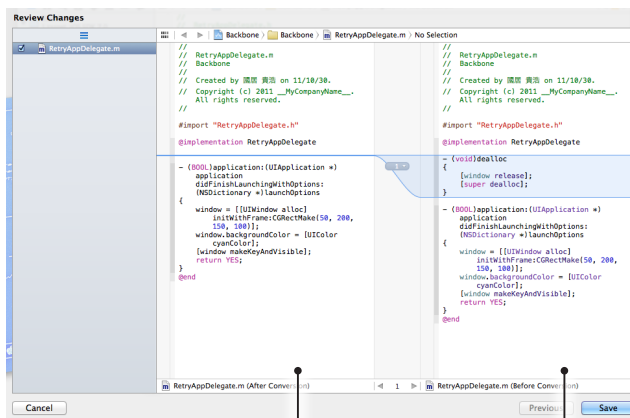
2 Check ボタンをクリックする

次に「これから ARC 対応への変更部分を示すので確認して欲しい」という画面が出るので Next ボタンをクリックします。



1 Next ボタンをクリックする

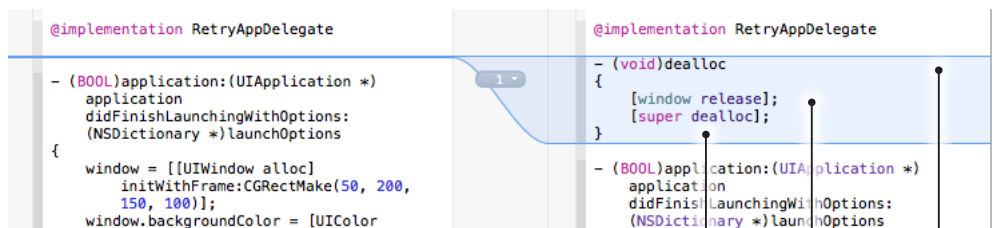
これでソースの変更部分が比較できる画面が表示されます。



ARC 対応へ変換した後のソース

変換前のソース

この画面を使って、ARC 対応でサンプルソースが、どのように変化するかを確認してください。



ARC 使用時は super 側の -dealloc を呼び出せないなので削除

コンパイラが面倒をみるので release の必要はないし、利用することもできないので削除

2つの処理が無くなると -dealloc メソッドそのものが必要なくなるので削除する

ARC 使用・非使用でのソースの違いを確認後は Cancel ボタンで終わらせてください。

ほとんどサンプルの `-dealloc` メソッドは、自分のインスタンス変数が示すオブジェクトの解放をおこない、`super` 側を呼び出すだけなので、まるごと削除となります。

ただし、中にはそれ以外の作業をおこなっている `-dealloc` メソッドもあり、例えば、サンプルプロジェクト 27/Rotate-5-view-transform の Rotate.xcodeproj だと次のようになります。

同じ `-dealloc` メソッドだが `[_window release]`、`[super dealloc]` 以外の処理は必要なので、今回は `dealloc` メソッド自体は残している

```
[[UIDevice currentDevice]
endGeneratingDeviceOrientationNotifications];
// 通知先を解除。
[[NSNotificationCenter defaultCenter] removeObserver:self];
}
// 通知先を解除。
[[NSNotificationCenter defaultCenter] removeObserver:self];
[_window release];
[super dealloc];
```

Convert to Objective-C ARC…メニューで示される差分を参考に、場合場合で対応していただく。

`autorelease` などは機械的に取りはぶくだけでかまわないでしょう。

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
self.window = [[UIWindow alloc] initWithFrame:[UIScreen
mainScreen] bounds];
// Override point for customization after application
launch.
self.window.backgroundColor = [UIColor whiteColor];
}
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
self.window = [[[UIWindow alloc] initWithFrame:[UIScreen
mainScreen] bounds]] autorelease];
// Override point for customization after application
launch.
self.window.backgroundColor = [UIColor whiteColor];
}
```

コンパイラが面倒をみるので `autorelease` の必要はないし、利用することもできないので、`autorelease` メッセージ部分だけ削除

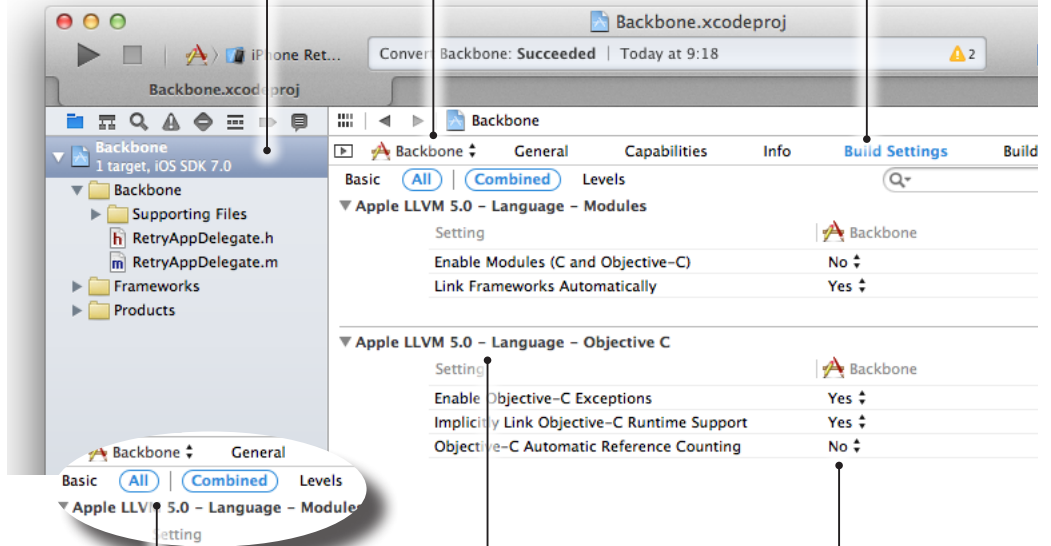
2、プロジェクトの設定を手動で変更し ARC を使用しないようにする

2 の場合は、プロジェクトの作成直後に、プロジェクトの Build Settings の Apple LLVM 5.0 - Language - Objective C の Objective-C Automatic Reference Counting の項目を NO にします。

1 プロジェクト項目選択

2 ターゲット選択

3 Build Settings タブ選択



4 All、Combined を選んでおく

5 Apple LLVM 5.0 - Language - Objective C
C が出るまでスクロールする

6 NO にする

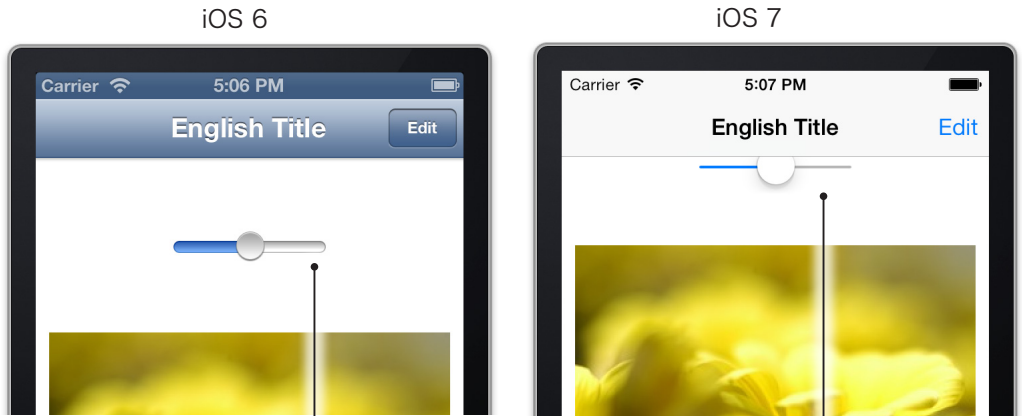
これで本にしたがった記述ができるようになります。

ただし、Xcode 側からテンプレートで提供されているソースは ARC 対応のままなので、その部分はダウンロードしたサンプルプロジェクト側のソースを参考に、自分で ARC 非使用の記述に書き直してください。

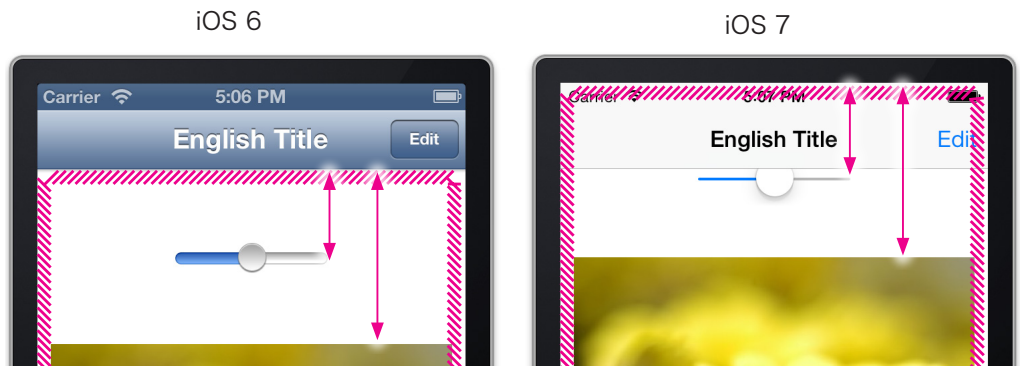
iOS7 対応について

iOS 7 では、ナビゲーションバーやステータスバーと内容部ビューの位置関係が見直され、内容部のビューがナビゲーションバーやステータスバーの下に潜り込む配置が標準となりました。

そのため、例えばサンプル：34/Localize-4-imageなどをRunすると、iOS 6とiOS 7でスライダの表示位置が大きく異なる事になります。



スライダのデザインも異なるが、表示位置も異なる。



内容部ビューがナビゲーションバーやステータスバーの下に潜り込む仕様が標準となったため

この問題を解決するには

サンプル：34/Localize-3-viewcontroller

サンプル：34/Localize-4-image

では LocalizeViewController.m の -viewDidLoad メソッドに次の処理を追加してください。

LocalizeViewController.m

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    if (floor(NSFoundationVersionNumber) > NSFoundationVersionNumber_iOS_6_1) {
        self.edgesForExtendedLayout = UIRectEdgeNone;
    }
    // Do any additional setup after loading the view from its nib.
}
```

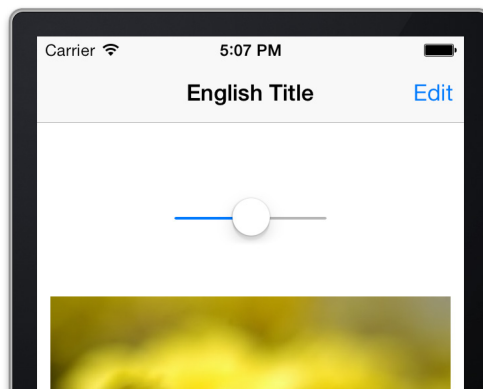
注意) Xcode 5 なので SDK には 7.0 を使う前提です。

最初の if 文は iOS のバージョンが 6.1 より大きいかどうかのチェックです。

大きい場合に、iOS 7 上で動作中とみなし、プロパティ `edgesForExtendedLayout` に `UIRectEdgeNone` を設定する事で、`LocalizeViewController` の管理している画面がナビゲーションバーの下に潜り込んでしまうのを防止しています。

詳細はクイックヘルプで、iOS 7 UI Transition Guide というガイドを参照してください。

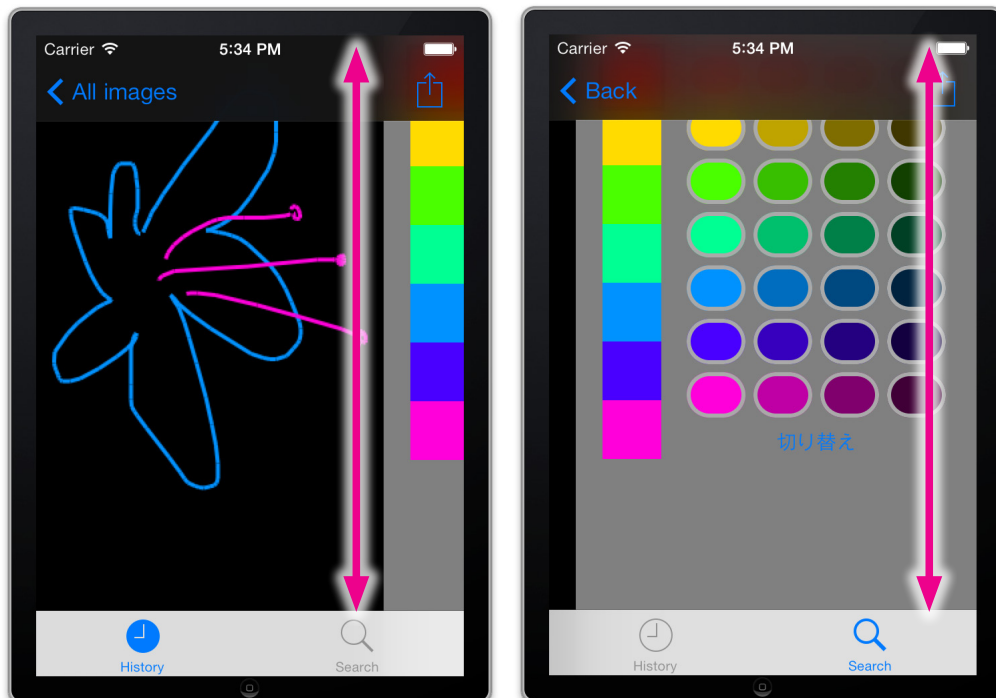
この処理の追加で、iOS 7 シミュレータで Run させると次のように、iOS 6 と同じ位置にスライダが表示されるようになります。



同じように

サンプル : 35/Graffiti

では次のような表示になってしまいます。



潜り込んでいる

この問題も、サンプル：34 と同じように対応します。

こちらは CanvasViewController.m の -viewDidLoad メソッドに同じ処理を追加してください。

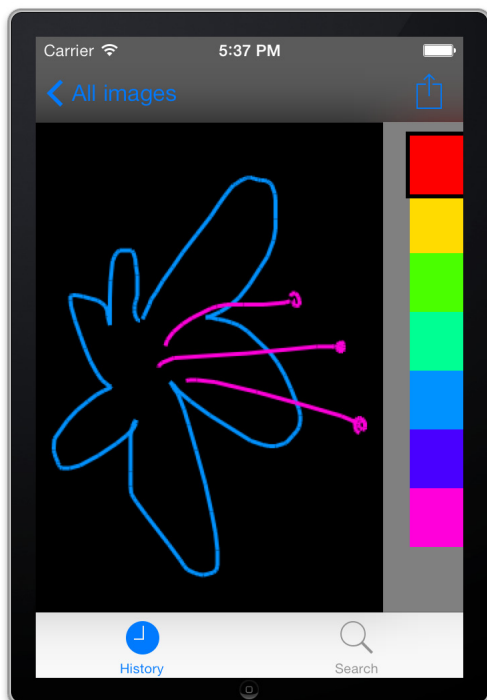
CanvasViewController.m

```

- (void)viewDidLoad
{
    [super viewDidLoad];
    if (floor(NSFoundationVersionNumber) > NSFoundationVersionNumber_iOS_6_1) {
        self.edgesForExtendedLayout = UIRectEdgeNone;
    }
    if (curtKeywords == nil)
        curtKeywords = [[NSMutableArray alloc] init];
    .
    .
    .
}

```

これで iOS 7 でも、正しく表示されます。



以上で Xcode 5 および iOS 7 対応の説明を終わります。
お疲れさまでした。