

iPhone与iPad应用开发课程 精通iOS开发

第八讲 应用程序设置

主讲人：关东升

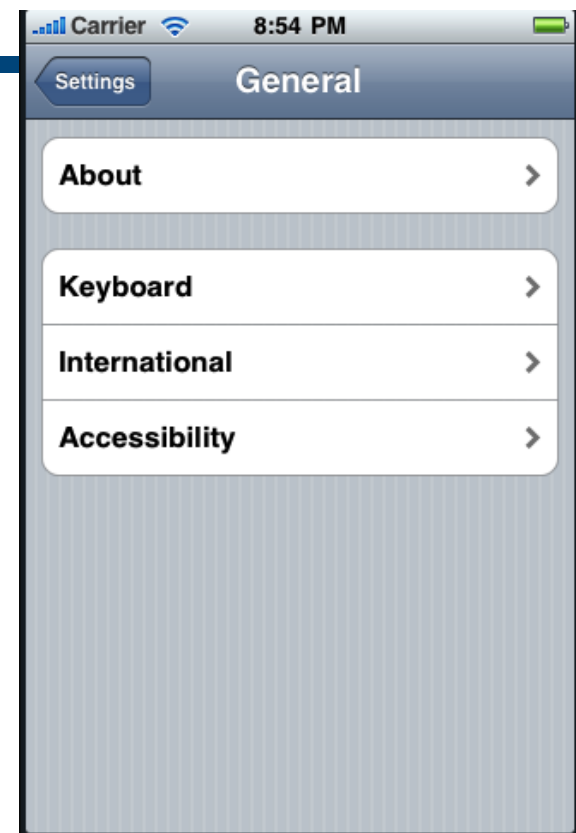
eorient@sina.com

主要知识点

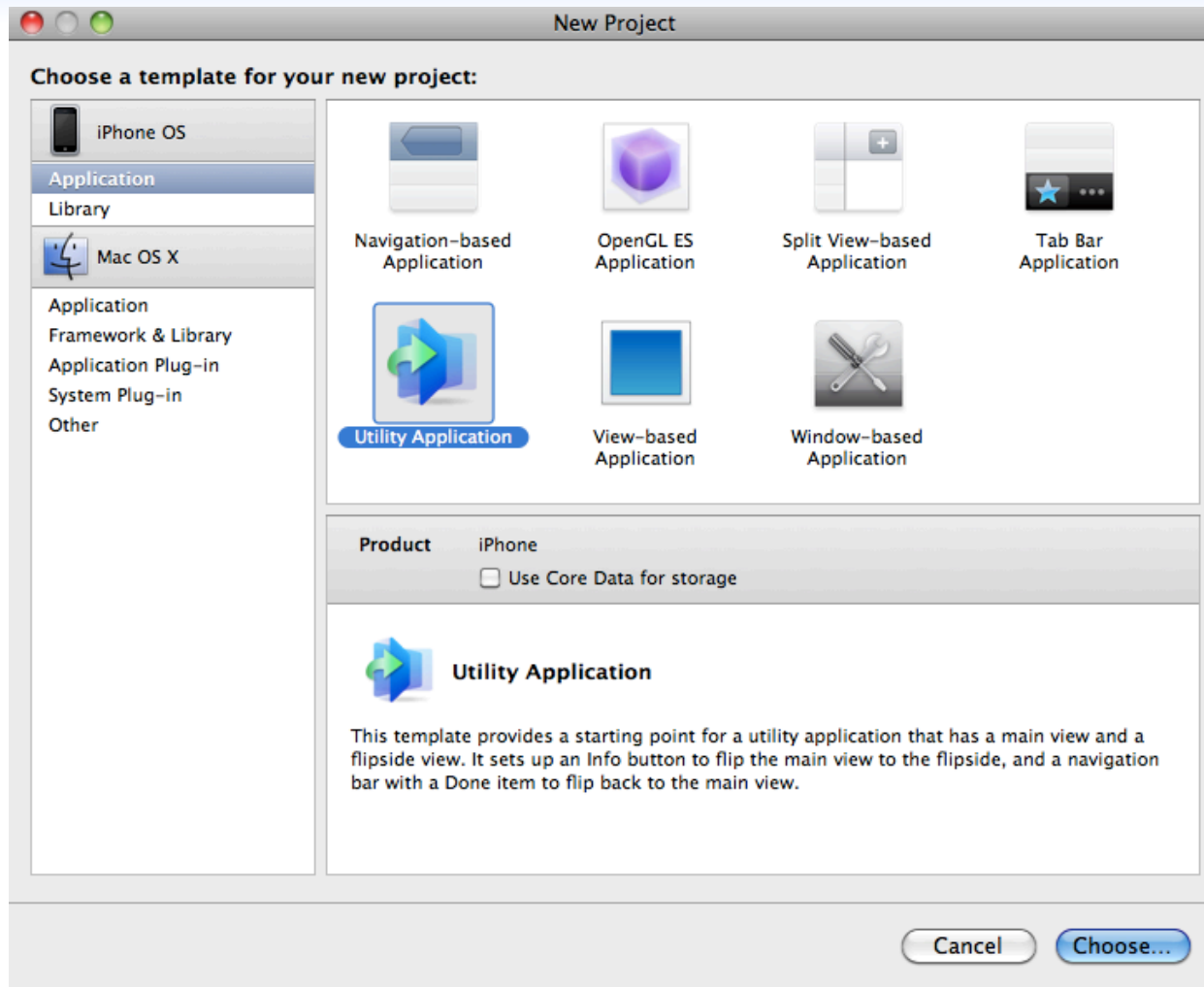
- ◆ 应用程序设置概念
- ◆ 创建设置束
- ◆ 设置项目种类
- ◆ 读取设置

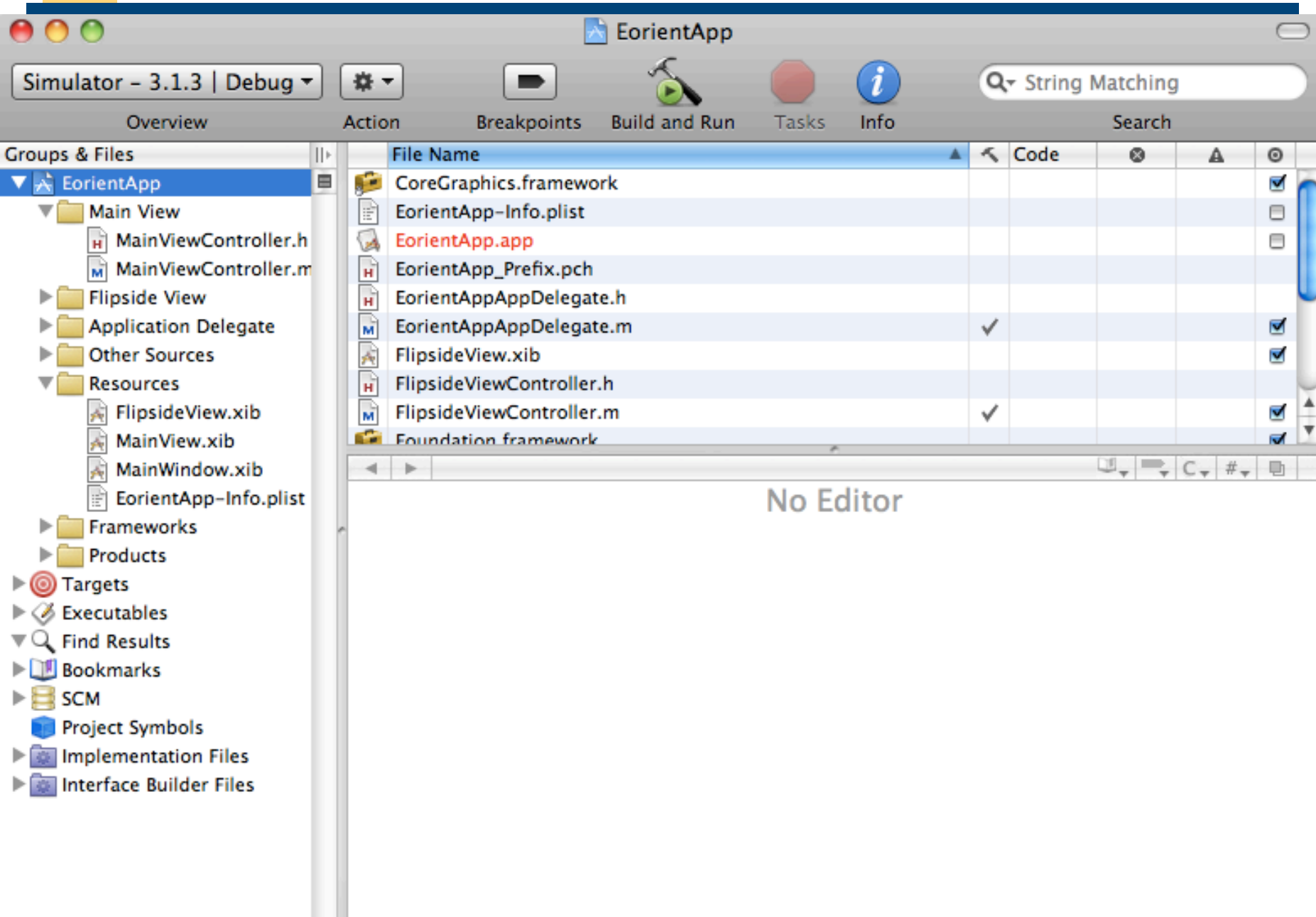
概述

- ◆ 应用程序设置（Application Settings）：在iOS和Mac OS很多软件中都有使用偏好（preferences），例如，在iOS中我们设置WiFi、运营商和壁纸等等。



创建工程

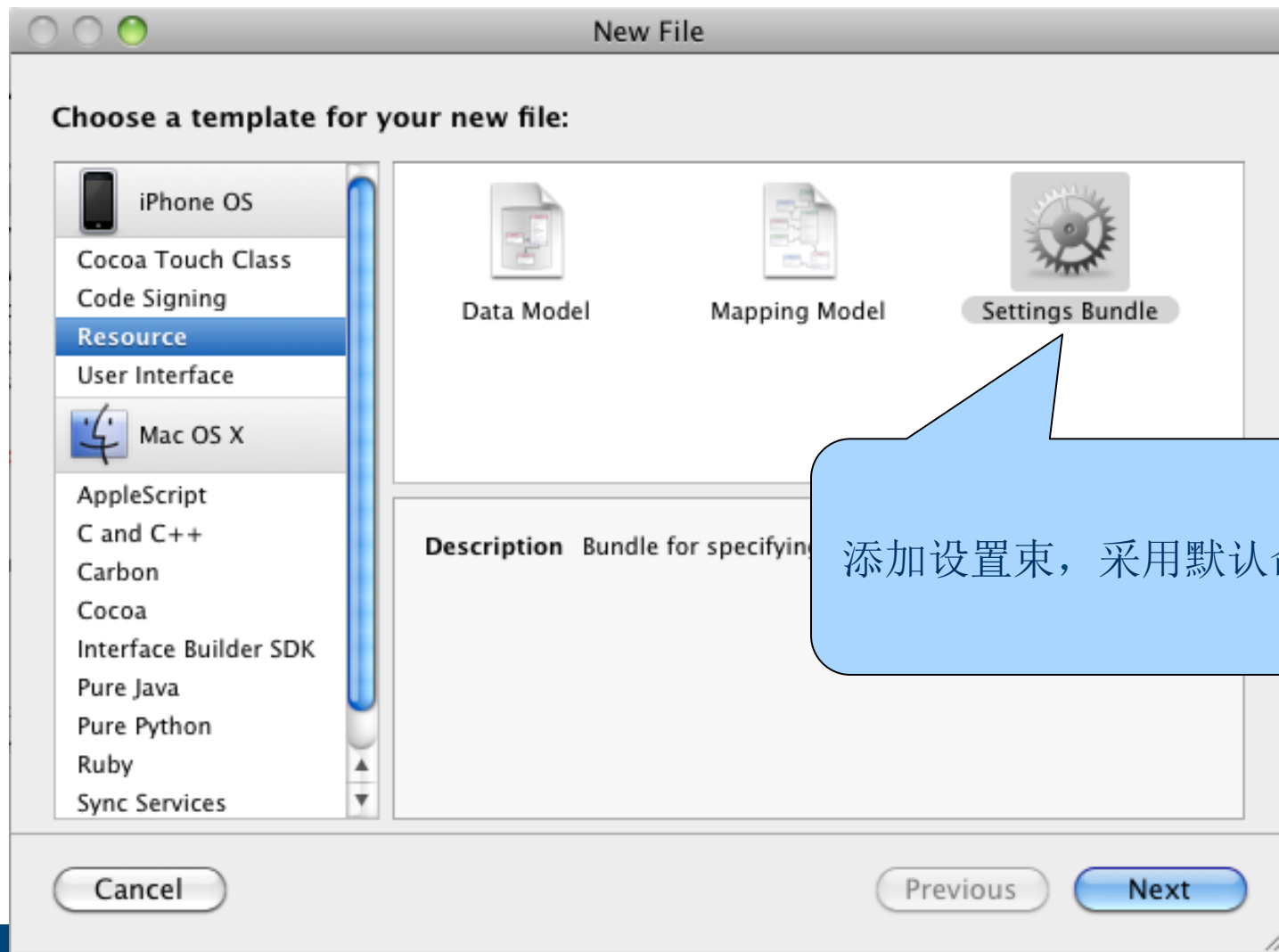




Settings Bundle（设置束）

- ◆ 一个应用程序的使用偏好，是存储在一个Root.list的XML文件中的，在这个文件中是与Settings Bundle关联起来的。可以在Root.list中指定设置项目的类型，可以有常规字段类型、划块、开关和子视图类型等等。

创建一个设置束



添加设置束，采用默认命名

Root.plist - EorientApp

Simulator - 3.1.2 | D... Overview Action Build and Go Tasks Info Search String Matching

Groups & Files

- EorientApp
 - Main View
 - Flipside View
 - Application Delegate
 - Other Sources
 - Resources
 - FlipsideView.xib
 - MainView.xib
 - MainWindow.xib
 - EorientApp-Info.plist
 - Settings.bundle
 - Root.plist
 - en.lproj
 - Root.strings
 - Frameworks
 - Products
 - EorientApp.app
 - Targets
 - Executables
 - Errors and Warnings
 - Find Results

File Name

- Root.plist
- Root.strings

确定后工程目录Root.plist文件，

用于应用程序的本地化

	Type	Value
Root	Dictionary	(2 items)
StringsTable	String	Root
PreferenceSpecifiers	Array	(4 items)
Item 1	Dictionary	(2 items)
Item 2	Dictionary	(8 items)
Item 3	Dictionary	(4 items)
Item 4	Dictionary	(7 items)

Debugging terminated. Succeeded

解释

- ◆ 上图中在Resource文件夹下添加设置束文件Settings.bundle(默认名称)。展开Settings.bundle文件夹：
 - Root.plist图标
 - en.lproj文件夹
- ◆ Root.plist是设置属性的属性列表
- ◆ en.lproj是本地化应用程序时使用的

Root.plist的PreferenceSpecifiers节点

The screenshot shows the Xcode IDE with the 'Root.plist' file open. The left sidebar displays the project structure, including 'EorientApp', 'Main View', 'Flipside View', 'Application Delegate', 'Other Sources', 'Resources', and 'Settings.bundle'. The main editor area shows the 'Root.plist' file with a table of preference specifiers.

Key	Type	Value
Root	Dictionary	(2 items)
StringsTable	String	Root
PreferenceSpecifiers	Array	(4 items)
Item 0	Dictionary	(2 items)
Item 1	Dictionary	(8 items)
Item 2	Dictionary	(4 items)
Item 3	Dictionary	(7 items)

Debugging terminated. Succeeded

运行程序



点击这个按钮
进入下个画面

这是运行这个工程出来的
第一个画面



点击这个按钮
进入下个画面



点击Settings这项
进入下个画面

接上图

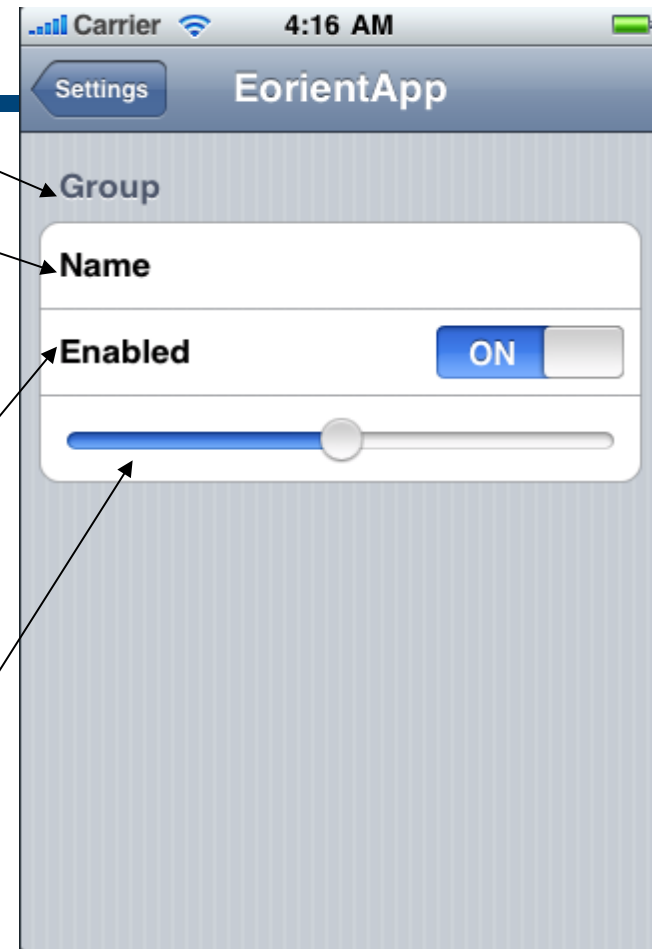


生成的设置束文件设置项目

- ◆ 从上面的运行的应用程序看，我们没有编写任何一行代码系统就已经帮我们创建好一些设置项目了。

Key	Type	Value
▼ Root	Dictionary	(2 items)
StringsTable	String	Root
▼ PreferenceSpecifiers	Array	(4 items)
▼ Item 1	Dictionary	(2 items)
Type	String	PSGroupSpecifie
Title	String	Group
▼ Item 2	Dictionary	(8 items)
Type	String	PSTextFieldSpec
Title	String	Name
Key	String	name_preferenc
DefaultValue	String	
IsSecure	Boolean	<input type="checkbox"/>
KeyboardType	String	Alphabet
AutocapitalizationType	String	None
AutocorrectionType	String	No
▼ Item 3	Dictionary	(4 items)
Type	String	PSToggleSwitchS
Title	String	Enabled
Key	String	enabled_preferenc
DefaultValue	Boolean	<input checked="" type="checkbox"/>
▼ Item 4	Dictionary	(7 items)
Type	String	PSSliderSpecifier
Key	String	slider_preferenc
DefaultValue	Number	0.5
MinimumValue	Number	0
MaximumValue	Number	1
MinimumValueImage	String	
MaximumValueImage	String	

Key	Type	Value
▼ Root	Dictionary	(2 items)
StringsTable	String	Root
▼ PreferenceSpecifiers	Array	(4 items)
▼ Item 1	Dictionary	(2 items)
Type	String	PSTextFieldSpecifier
Title	String	Group
▼ Item 2	Dictionary	(8 items)
Type	String	PSTextFieldSpecifier
Title	String	Name
Key	String	name_preference
DefaultValue	String	
IsSecure	Boolean	<input type="checkbox"/>
KeyboardType	String	Alphabet
AutocapitalizationType	String	None
AutocorrectionType	String	No
▼ Item 3	Dictionary	(4 items)
Type	String	PSToggleSwitchSpecifier
Title	String	Enabled
Key	String	enabled_preference
DefaultValue	Boolean	<input checked="" type="checkbox"/>
▼ Item 4	Dictionary	(7 items)
Type	String	PSSliderSpecifier
Key	String	slider_preference
DefaultValue	Number	0.5
MinimumValue	Number	0
MaximumValue	Number	1
MinimumValueImage	String	
MaximumValueImage	String	



PSGroupSpecifier

- ◆ 设置项目Type为PSGroupSpecifier用于指示该项目是一个新分组的开始，其后的每个项目都将是此分组的一部分，直到一个Type值为PSGroupSpecifier的项目之前。
- ◆ 在本例子中我们还要设置Title为Group，有多个Group时候Title不能重复。

PSTextFieldSpecifier

- ◆ 设置项目Type为PSTextFieldSpecifier是指示该项目是一个特定的使用偏好字段类型。其它项目的说明：

键	值	说明
Type	PSTextFieldSpecifier	字段类型
Title	Name	项目的标题
Key	name_preference	项目的key
DefaultValue		默认值
IsSecure	false	是否为安全字段
KeyboardType	Alphabet	字母键盘
AutocapitalizationType	None	是否自动大写
AutocorrectionType	No	是否拼写纠正

PSToggleSwitchSpecifier

- ◆ 设置项目Type为PSToggleSwitchSpecifier是指示该项目是一个特定的使用偏好开关类型。其它项目的说明：

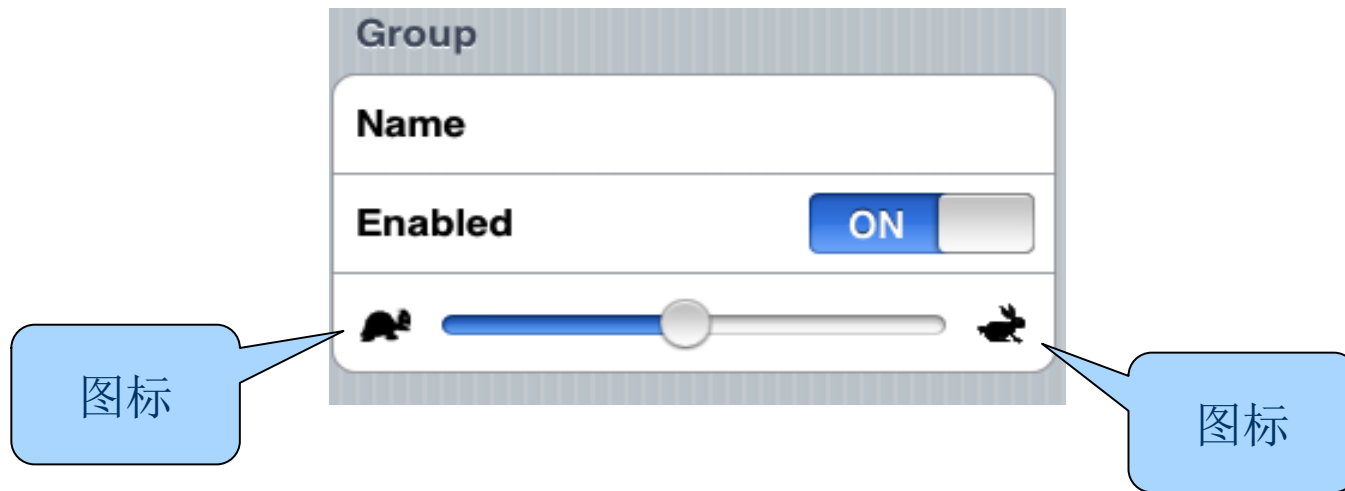
键	值	说明
Type	PSToggleSwitchSpecifier	开关类型
Title	Enabled	项目的标题
Key	Enabled	项目的key
DefaultValue	true	默认值

PSSliderSpecifier

- ◆ 设置项目Type为PSSliderSpecifier是指示该项目是一个特定的使用偏好滑块类型。其它项目的说明：

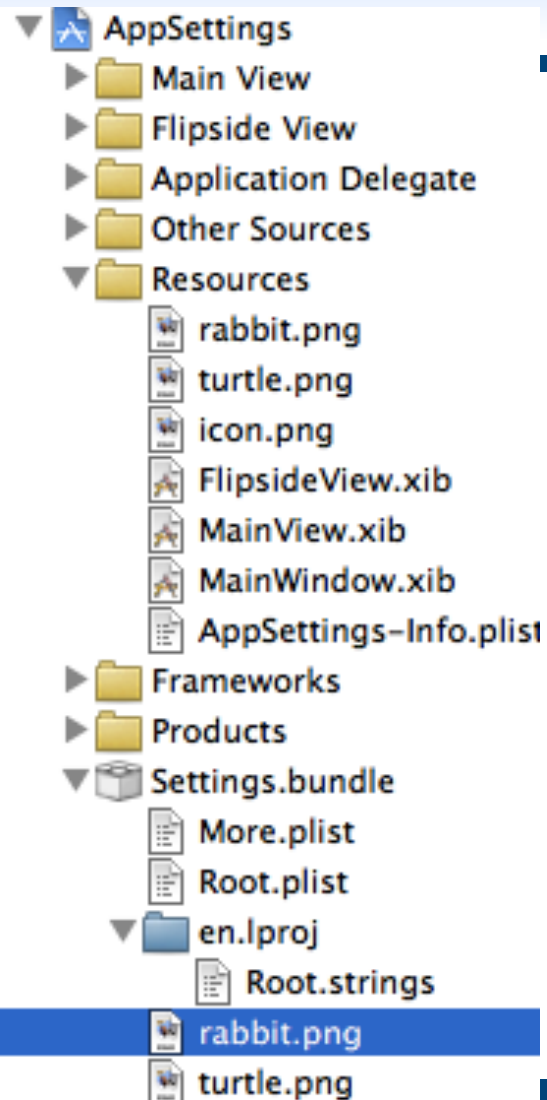
键	值	说明
Type	PSSliderSpecifier	开关类型
Key	slider_preference	项目的key
DefaultValue	0.5	默认值
MinimumValue	0	最小值
MaximumValue	1	最大值
MinimumValueImage	turtle.png	最小值图片
MaximumValueImage	rabbit.png	最大值图片

为滑块添加最大最小值图片

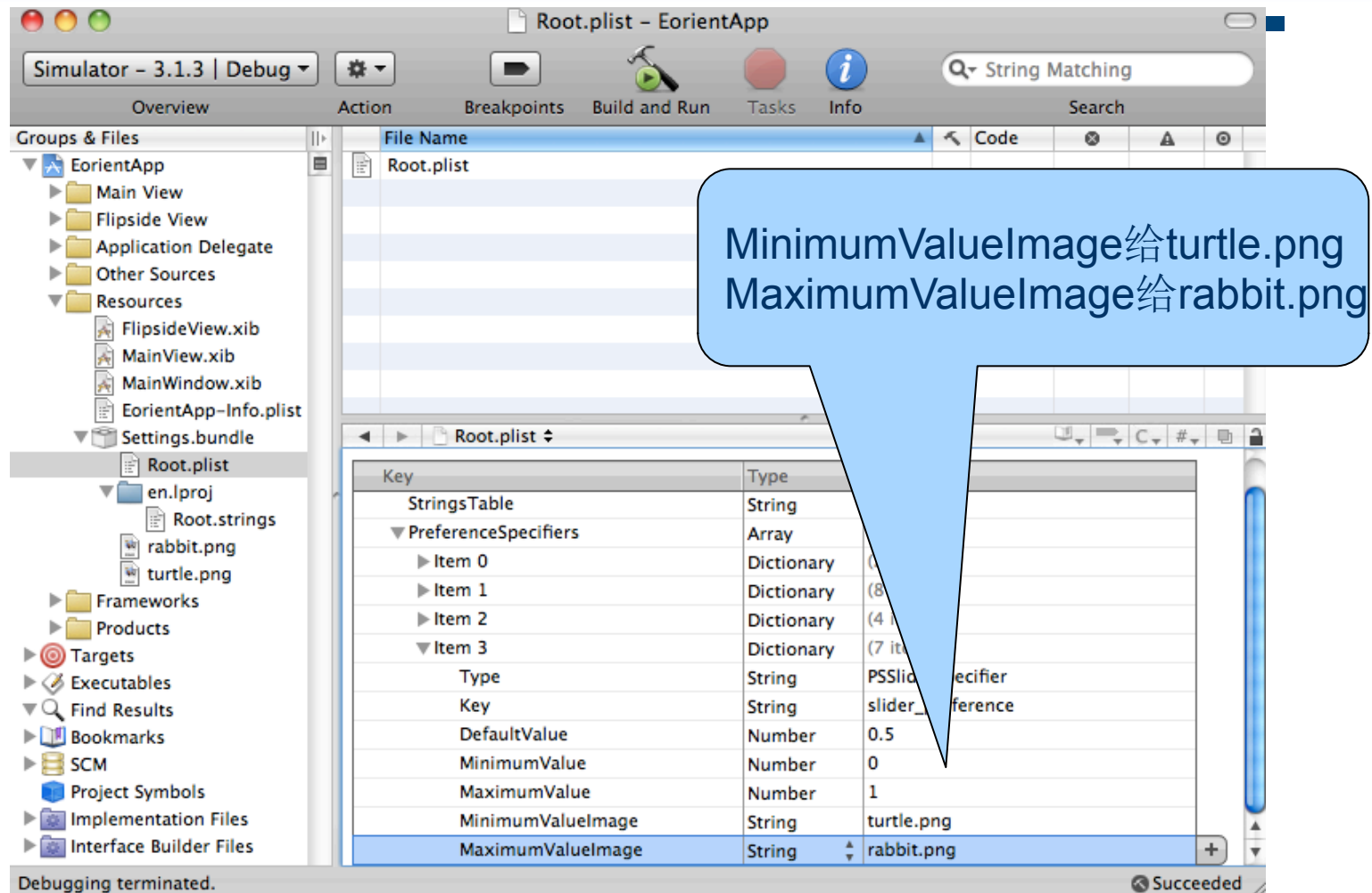


- ◆ **Show Package Contents**来访问束的内容。这将打开一个新的窗口。将图标文件复制到此文件夹中。这样束就能找到这两个图标文件了。

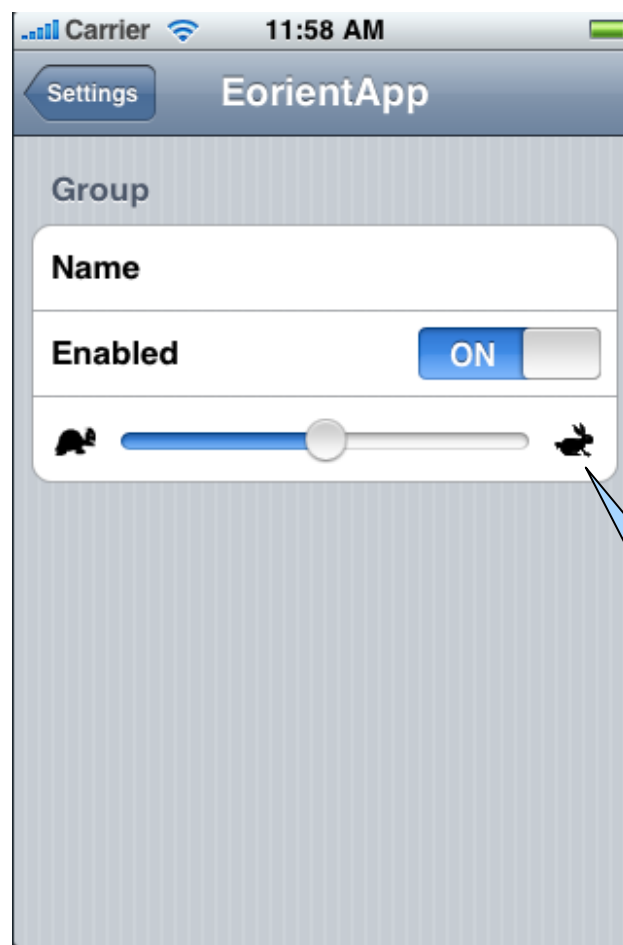
添加最大最小值图片文件结构



具体往束里设置图片：



运行结果



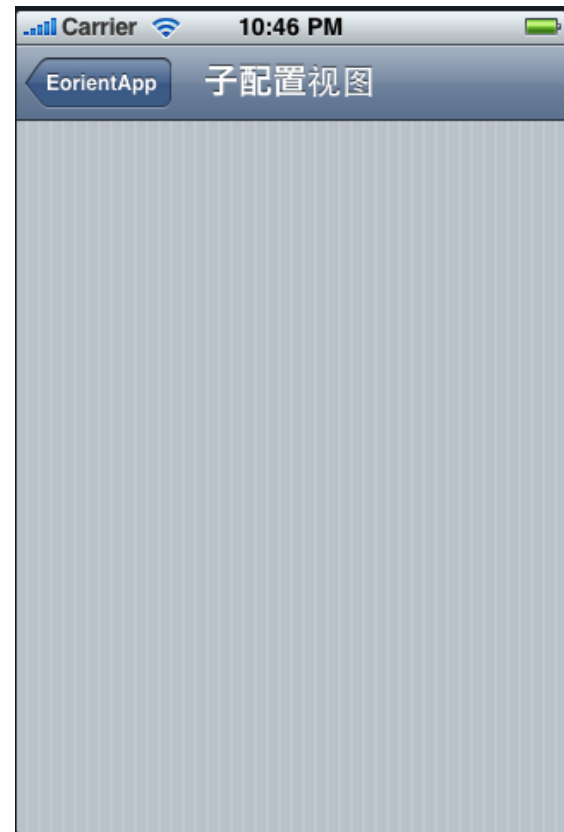
加了图标的

添加子设置视图

Key	Type	Value
▼ Root	Dictionary ▲	(2 items)
StringsTable	String	Root
▼ PreferenceSpecifiers	Array	(5 items)
▶ Item 0	Dictionary	(2 items)
▶ Item 1	Dictionary	(8 items)
▶ Item 2	Dictionary	(4 items)
▶ Item 3	Dictionary	(7 items)
▼ Item 4	Dictionary	(3 items)
Type	String	PSChildPaneSpecifier
Title	String	子设置视图
File	String	Shades

添加完子视图，运行程序：

- 其中File是指导子视图配置设置束（Shades.plist）



读取应用程序中设置

- ◆ 我们使用NSUserDefaults类读取用户设置，通常使用
- ◆ `NSUserDefaults* defaults = [NSUserDefaults standardUserDefaults];`
- ◆ NSUserDefaults获得值方法有：
- ◆ `arrayForKey:` `boolForKey:` `dataForKey:`
- ◆ `dictionaryForKey:` `floatForKey:` `integerForKey:`
`objectForKey:` `stringArrayForKey:` `stringForKey:`



MainViewController.h

```
#import "FlipsideViewController.h"

@interface MainViewController : UIViewController
<FlipsideViewControllerDelegate> {
    UILabel *username;
}
@property (nonatomic, retain) IBOutlet UILabel *username;

- (IBAction)showInfo;

@end
```

MainViewController.m

```
- (void)viewDidUnload {  
    self.username = nil;  
}  
- (void)viewDidAppear:(BOOL)animated {  
    NSUserDefaults* defaults = [NSUserDefaults standardUserDefaults];  
    username.text = [defaults objectForKey:@"name_preference"];  
    [super viewDidAppear:YES];  
}
```

FlipsideViewController.h

```
@protocol FlipsideViewControllerDelegate;

@interface FlipsideViewController : UIViewController {
    id <FlipsideViewControllerDelegate> delegate;
    UILabel *myPSToggleSwitchSpecifier;
    UILabel *myPSSliderSpecifier;
}

@property (nonatomic, retain) IBOutlet UILabel *myPSToggleSwitchSpecifier;
@property (nonatomic, retain) IBOutlet UILabel *myPSSliderSpecifier;
@property (nonatomic, assign) id <FlipsideViewControllerDelegate> delegate;
- (IBAction)done;

@end

@protocol FlipsideViewControllerDelegate
- (void)flipsideViewControllerDidFinish:(FlipsideViewController *)controller;
@end
```

FlipsideViewController.m

```
#import "FlipsideViewController.h"
```

```
@implementation FlipsideViewController
```

```
@synthesize delegate;
```

```
@synthesize myPSToggleSwitchSpecifier;
```

```
@synthesize myPSSliderSpecifier;
```

```
- (void)viewDidLoad {
```

```
    NSUserDefaults* defaults = [NSUserDefaults standardUserDefaults];
```

```
    NSString *text = [[NSString alloc] initWithFormat: @"%f",  
                                                         [defaults floatForKey:@"slider_preference"]];
```

```
    myPSSliderSpecifier.text = text;
```

```
    [text release];
```

```
    myPSToggleSwitchSpecifier.text = [defaults stringForKey:@"enabled_preference"];
```

```
    [super viewDidLoad];
```

```
}
```

FlipsideViewController.m

```
- (IBAction)done {  
    [self.delegate flipsideViewControllerDidFinish:self];  
}  
  
- (void)viewDidUnload {  
    self.myPSSliderSpecifier = nil;  
    self.myPSToggleSwitchSpecifier = nil;  
}  
  
- (void)dealloc {  
    [myPSSliderSpecifier release];  
    [myPSToggleSwitchSpecifier release];  
    [super dealloc];  
}
```

@end

运行

