

zhaozhy55555的专栏

目录视图

摘要视图

RSS 订阅

个人资料



zhaozhy55555

访问： 153721次

积分： 2100

等级：

BLOG > 5

排名： 第10401名

原创： 39篇 转载： 62篇

译文： 0篇 评论： 25条

文章搜索

文章分类

- 编程语言 (4)
- TD-SCDMA (2)
- 天上的星-亮晶晶 (21)
- IOS应用 (58)
- English (0)
- 操作系统-linux (3)
- 数据结构与算法 (1)
- 程序架构 (0)
- 网络协议 (2)
- 资料翻译 (0)
- 文章收藏 (0)
- couchbase (1)
- 安全 (0)

文章存档

- 2015年03月 (1)
- 2013年12月 (2)
- 2013年10月 (3)
- 2013年03月 (5)

想听课？来发话题吧

云服务器使用初体验~获奖公告

技术干货、还免费？来这儿

有奖试读—增长黑客，创业

技”

XCode工程中 Project 和 Targets区别

分类： IOS应用

2013-01-30 17:40

9937人阅读

评论

XCode工程中 Targets讲解是本文要介绍的内容，相信很多人都注意到XCode中，有个Target的概念。这在很多地方都有所体现，比如打开一个工程后，左侧的列表中有Targets一项，而在工程界面的顶部菜单中，project里面也有多个涉及到Target的项目，那么这个Target到底是什么呢？

Apple的人是这样说的：“Targets that define the products to build. A target organizes the files and instructions needed to build a product into a sequence of build actions that can be taken.”

简单的理解的话，可以认为一个target对应一个新的product(基于同一份代码的情况下)。但都一份代码了，弄个新product做啥呢？折腾这个有意思么？

其实这不是单纯的瞎折腾，虽然代码是同一份，但编译设置(比如编译条件)，以及包含的资源文件却可以有很大的差别。于是即使同一份代码，产出的product也可能大不相同。

我们来举几个典型的应用多Targets的情况吧，比如完整版和lite版；比如同一个游戏的20关，30关，50关版；再或者比如同一个游戏换些资源和名字就当新游戏卖的(喂喂，你在教些什么...)

Targets之间，什么相同，什么不同

既然是利用同一份代码产出不同的product，那么到底不同Target之间存在着什么样的差异呢？

要解释这个问题，我们就要来看看一个Target指定了哪些内容。

从XCode左侧的列表中，我们可以看到一个Target包含了Copy Bundle Resources, Compile Sources, Link Binary With Libraries。其中

Copy Bundle Resources 是指生成的product的.app内将包含哪些资源文件

Compile Sources 是指将有哪些源代码被编译

Link Binary With Libraries 是指编译过程中会引用哪些库文件

通过Copy Bundle Resources中内容的不同设置，我们可以让不同的product包含不同的资源，包括程序的主图标等，而不是把XCode的工程中列出的资源一股脑的包含进去。

而这还不是一个target所指定的全部内容。每个target可以使用一个独立，不同的Info.plist文件。

我们都知道，这个Info.plist文件内定义了一个iPhone项目的很多关键性内容，比如程序名称，最终生成product的全局唯一id等等。

而且不同的target还可以定义完整的差异化的编译设置，从简单的调整优化选项，到增加条件编译所使用的编译条件，以至于所使用的base SDK都可以差异化指定。

2013年02月 (13)

展开

阅读排行

IOS 发布程序到AppStore (9984)

xcode 代码自动对齐 快捷键 (9945)

XCode工程中 Project 和 (9921)

设置socket缓冲区大小 (9415)

RLC三种传输模式 (8242)

IOS谓词--NSPredicate (8005)

U盘装系统 (7072)

CALayer (4076)

ios 弱引用 (3867)

ARM指令集和X86指令集 (3465)

评论排行

xcode 代码自动对齐 快捷键 (14)

RLC三种传输模式 (4)

设置socket缓冲区大小 (3)

如何隐藏win32 console (1)

ARQ自动重传请求 (1)

XCode: 兼容ARC和非- (1)

IOS谓词--NSPredicate (1)

退避算法 (1)

layoutSubviews调用时机 (0)

BSD-GPL (0)

推荐文章

*在R中使用支持向量机 (SVM) 进行数据挖掘 (上)

* 你不再需要动态网页——编辑-发布-开发分离

*Android性能优化之使用线程池处理异步任务

*通过Ajax的方式执行GP服务

*编译器架构的王者LLVM——(6) 多遍翻译的宏翻译系统

* 【笨木头Unity】入门之旅010 (完结) : Demo之四处找死 (五) _UI

最新评论

xcode 代码自动对齐 快捷键 dylanli1986: command + a, control+I岂不更好?

xcode 代码自动对齐 快捷键 w1667: 哈哈, 好屌~~~

xcode 代码自动对齐 快捷键 林深己莫知: 好像也没有什么不对的 23333

xcode 代码自动对齐 快捷键 绝对Qiang: 哈哈, 也是人才

设置socket缓冲区大小 我还aini: 这个大小有上限吗?

xcode 代码自动对齐 快捷键 aimsgmiss: 2b

IOS谓词--NSPredicate Demon_Study: 感谢楼主分享,

创建第二个Target

为什么是第二个? 因为第一个就是创建好工程后的默认Target呀! (废话这么多, 拖走...)

创建target有多种方法, 我们可以从现有的target上复制出一份, 然后略加改动, 也可以完全新建一个target出来. 但其实说穿了, 两个方法大同小异

首先我们来看看利用复制的方法创建target

利用复制创建target

我们在XCode左侧的列表中, 展开 Targets 项, 在现有的target上, 右键选择 "Duplicate", 或者选中现有target后, 在顶部菜单的Edit内选择"Duplicate"也可以.

此时我们就得到了一个新的target, 而在Resource里面也会得到一个 xxxx copy.plist. 这个新的target与原有的target是完全一致的, 余下的就是一些差异化的修改, 这个我们后面再说

创建全新的target

类似复制的方法, 我们可以在左侧的列表中很多地方按下右键菜单, 都可以看到Add中会有"New Target..."一项, 而在工程顶部菜单的Project内, 也可以看到这个"New Target..."的身影.

点击后, 首先会让你选择target的类型, 既然我一直所指的是程序本身, 那么自然选择Application了(至于其他的嘛, 有兴趣的自己研究吧, 比如我们可以把程序中的部分提取成一个Static Library).

Next后, 会让你输入一个新的Target的名字, 而不像复制的方法中, 默认生成 xxxxx copy这样的target名.

但是这样生成出的Target几乎是空的. Copy Bundle Resources, Compile Sources, Link Binary With Libraries里面都没有任何内容. 编译设置也是完全原始的状态.

可以通过拖拽内容到这些target的设置中, 以及调整编译选项来完成Target的配置。

Target中部分内容的修改方法

其实这段的部分内容, 在非多Targets的工程中也可能会有用到.

由于修改基本都是在工程/编译设置中完成, 因此没有特殊情况. 就不再声明了, 打开target对应的工程/编译设置的方法可以采用在该target上右键, 选择get info来做到.

生成的product名称的修改: Packing段内的Product Name一项

Info.plist文件名: Packing段内的Info.plist File一项, 比如复制出来的target觉得那个xxxxx copy.plist太傻就可以在这里改

条件编译: 增加一个User-Defined Setting(Target "xxxx" Info的build页的左下角那个齿轮中可以看到这个内容), 在Other C Flag里面填入, 比如要定义一个叫做LITE_VERSION的define值, 我们可以写上 "-DLITE_VERSION" 或 "-DLITE_VERSION=1". 那么在程序中就可以用

```
#if defined(LITE_VERSION)
#else
#endif
```

这样的条件编译来部分差异化代码了

也许有些朋友记得我在代码区贴过的检测破解版的代码, 其中有一种检测方法就是看info.plist是文本还是二进制的, 那么我们能否建议一个模拟破解的target, 直接生成文本的info.plist以便测试呢?

当然可以, 在packing段内, 有一项叫"Info.plist Output Encoding", 默认值是Binary, 我们只要选成xml, 那么生成出的product.app内的info.plist就直接是文本样式的了.

另外, 向Copy Bundle Resources, Compile Sources, Link Binary With Libraries内添加/删除文件, 可以在要改动的文

太强大了 以前谓词只接触皮毛，收藏起来，以后肯定会用的到！！！！

xcode 代码自动对齐 快捷键 QQ809568546: 神人啊,谢谢

xcode 代码自动对齐 快捷键 fortunely2: 经实际操作验证验证，按照lz说得操作之后，代码可以自动对齐。

xcode 代码自动对齐 快捷键 zyunn: 额~能达到目的不就好了，这个方法虽然不是正统，但是确实有效~

件上, 选择get info, 并且切换到Target页, 勾选要引用这个文件的target即可. 比如icon.png可以指定给默认target, 而icon_lite.png指定给lite verion的target

- 上一篇 XCode: 兼容ARC和non-ARC
- 下一篇 深入浅出 iOS 之多线程 NSThread

顶

1

踩

0

主题推荐

xcode

color

rgb

猜你在找



查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

apttech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved