

fengsh998的专栏

blog因你而精彩。

☰ 目录视图

☰ 摘要视图

RSS 订阅

个人资料



fengsh998

访问：929907次

积分：8786

等级：

BLOG 5

排名：第1287名

原创：94篇

转载：50篇

译文：5篇

评论：178条

联系方式

QQ:19985430(满)/3031702

fengsh998@163.com

文章搜索

文章分类

- iphone (65)
- mac os (4)
- 网络 (0)
- 网络设置 (1)
- DOS (4)
- Delphi (7)
- Delphi 线程 (3)
- android 资料 (4)
- Object-C (3)
- 设置 (5)

【专家问答】阿里陈康贤：探讨大型网站之架构

走进VR开发世界——我们离开发一款VR大作还有多远？

【免费公开课】Python黑客编程之局域网嗅探和敏感信息探测

初探swift语言的学习笔记九(OC与Swift混编)

标签：

object-c

swift

混编

OC

2014-06-25 22:26

41399人阅读

评论(5)

收藏

举报

☰ 分类：

swift (12)

 ▼

❗ 版权声明：本文为博主原创文章，未经博主允许不得转载。

作者：fengsh998

原文地址：<http://blog.csdn.net/fengsh998/article/details/34440159>

转载请注明出处

如果觉得文章对你有所帮助，请通过留言或关注微信公众帐号fengsh998来支持我，谢谢！

swift 语言出来后，可能新的项目直接使用swift来开发，但可能在过程中会遇到一些情况，某些已用OC写好的类或封装好的模块，不想再在swift 中再写一次，哪就使用混编。这个在 I O S 8中是允许的。

先中简单的入手，先研究在同一个工程目录下混合使用的情况。

为了演示。先准备两个类

第一个是swift语言写的类,文件名为 act.swift

```
import Foundation

class Act : NSObject
{
    func hasAct(tag:Int) -> String
    {
        switch (tag)
        {
            case 1:return "Movie"
            case 2:return "CCTV"
            case 3:return "Sport TV"
            default:return "Area TV"
        }
    }

    init()
```

角色 (5)  
设计模式 (5)  
安全 (6)  
UML (2)  
C++ (21)  
gcc/g++ (11)  
架构 (5)  
编码 (7)  
WCF(C#) (3)  
Linux (2)  
医学 (1)  
swift (13)

#### 文章存档

2015年05月 (2)  
2014年07月 (1)  
2014年06月 (13)  
2014年05月 (1)  
2014年03月 (1)  
2014年01月 (2)  
2013年11月 (1)  
2013年09月 (2)  
2013年08月 (2)  
2013年06月 (5)  
2013年05月 (2)  
2013年04月 (22)  
2013年03月 (18)  
2013年02月 (9)  
2013年01月 (4)  
2012年12月 (9)  
2012年11月 (16)  
2012年10月 (34)  
2012年09月 (15)  
2012年08月 (9)  
2011年12月 (1)  
2011年03月 (2)  
2010年11月 (1)  
2010年07月 (1)

#### 阅读排行

IOS SEL (@selector) 原 (60914)  
IOS CoreText.framework (43441)  
初探swift语言的学习笔记 (41289)  
写给喜欢用Block的朋友(i (39708)  
ios设备抓包方法 (34833)  
IOS 世界标准时间UTC / (34679)  
IOS CoreText.framework (29132)  
WINHTTP的API接口说明 (21739)  
IOS Emoji 编码打印 (19317)  
ios 防止按钮快速点击造 (18061)

#### 评论排行

基于智能手机的加速计进 (20)

```
{
    println("act constructor is called.")
}

deinit
{
    println("act destroyed is called.")
}
}
```

第二个是用OC写的类 头文件为OCChannel.h,实现文件为OCChannel.m  
头文件

```
#import <Foundation/Foundation.h>

@interface OCChannel : NSObject

@property (nonatomic,retain) NSString *ChannelName;

- (NSString *)ChannelChange:(NSInteger) channels;

@end
```

实现文件

```
#import "OCChannel.h"
#import "SwiftModule-swift.h"

@interface OCChannel()
{
    Act *act; //swift的类
}
@end

@implementation OCChannel

- (id)init
{
    if (self = [super init]) {
        NSLog(@"OC Constructor is called.");
        //使用Swift类
        act = [[Act alloc] init];
    }
    return self;
}

- (void)dealloc
{
    NSLog(@"OC Destroyed is called.");
}
```

IOS CoreText.framework (18)  
写给喜欢用Block的朋友(i (16)  
IOS CoreText.framework (13)  
详释(常见UITextView 输, (10)  
IOS CoreText.framework (9)  
IOS SEL (@selector) 原 (9)  
自荐亲自写的一套http访 (7)  
UITextView的动态适应高 (6)  
初探swift语言的学习笔记 (6)

#### 推荐文章

\*EventBus的使用与深入学习  
\*Android 拍照、选择图片并裁剪  
\*spark性能调优：开发调优  
\*浅谈android中图片处理之色彩特效处理ColorMatrix(三)  
\*neutron-server的启动流程(一)  
\* iOS 网络资源汇总之动画

#### 最新评论

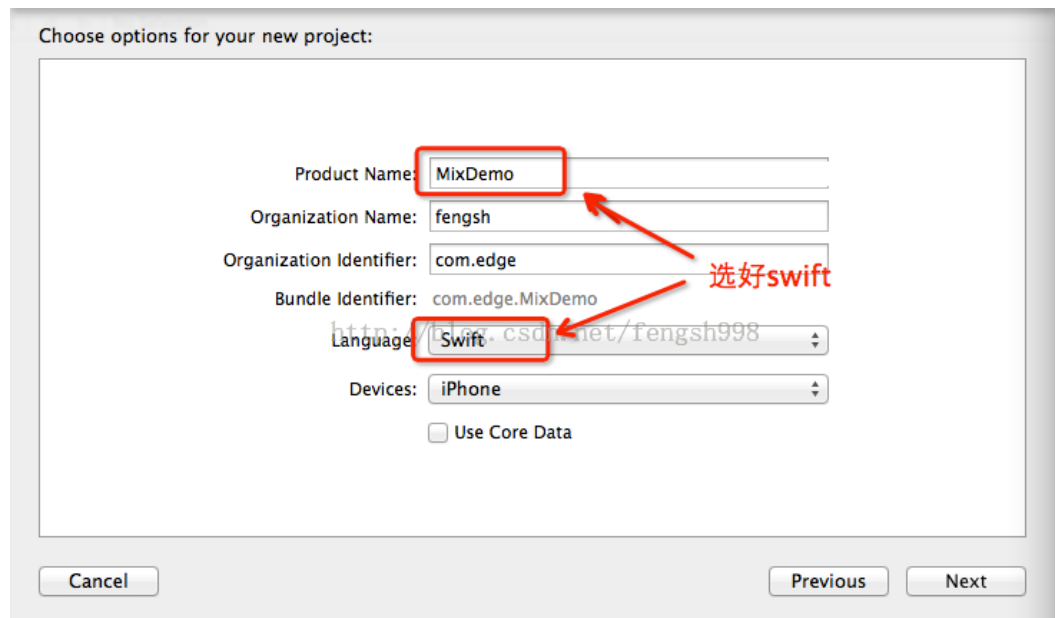
写给喜欢用Block的朋友(ios Bloc  
chenhening: 有钻研精神, 思考  
后的劳动成果。  
基于智能手机的加速计进行计步  
baidu\_34203140: 求来份demo,  
1007352884@qq.com  
基于智能手机的加速计进行计步  
qq\_18670721: 群主可以吧dome  
飞来一份看看么?  
578391264@qq.com 谢谢!  
详释(常见UITextView 输入之字数  
sinat\_33391833: 有个问题, 就  
是在已经字数全满的情况下, 用  
系统键盘的联想文字插入的时  
候, 是可以插入成功的, 然后光  
标会移...  
基于智能手机的加速计进行计步  
baidu\_33756690: 博主你好, 可  
以把这个代码发给我学习一下  
么, 万分感谢。邮箱:  
15035877332@163.com  
UITextView的动态适应高度处理  
ztq1988: self.autoAdpterRect 中  
的autoAdpterRect 指的是什么?  
详释(常见UITextView 输入之字数  
tinysail: 好  
详释(常见UITextView 输入之字数  
tinysail: 好  
IOS CoreText.framework --- 段落  
qq\_26653181: 如何可以改变字  
体颜色和大小呢??  
初探swift语言的学习笔记二(可选  
wushuirenjia: 很棒, , 谢谢

```
[[act release]; //ARC not use  
[[super dealloc]; //ARC not use  
}  
  
- (NSString *)ChannelChange:(NSInteger) channels  
{  
    return [act hasAct:channels];  
}  
  
@end
```

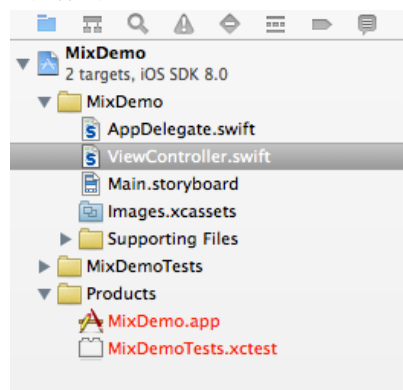
这个OCChannel为中引用了swift 写的类Act 。主要是为了演示在同一个工程项目里, swift类调用OC,同时OC类也调用Swift。从而形成一种混合编写的模式。

下面是具体步骤:

1.新建一个Swift工程: 我这里工程名为MixDemo



建好后工程:



2.就是分别引入前面的两个类, 咱先一个来。因为建的是Swift, 所以, 咱先以Swift工程中引用OC文件进行一次混编

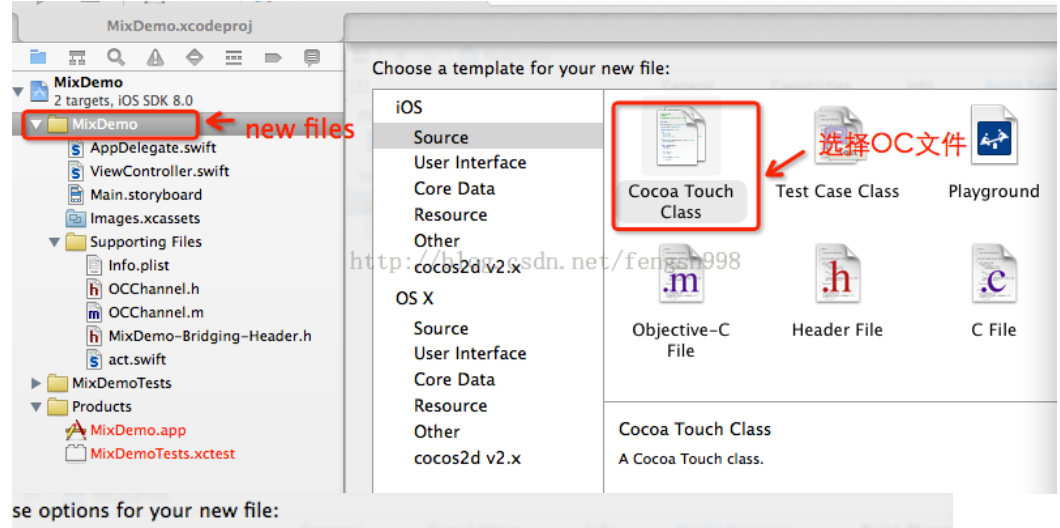
#### Swift中使用OC

首先Swift中不再使用头文件和.m文件的方式了。所以也不需要import ""来导入头文件。哪swift 如何能访问到O C的类声明的。

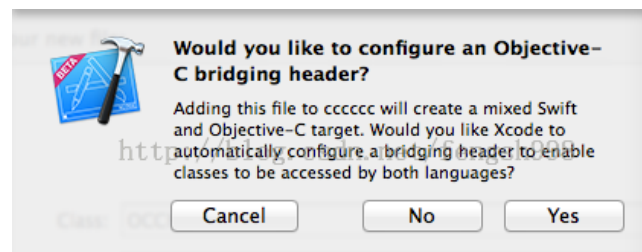
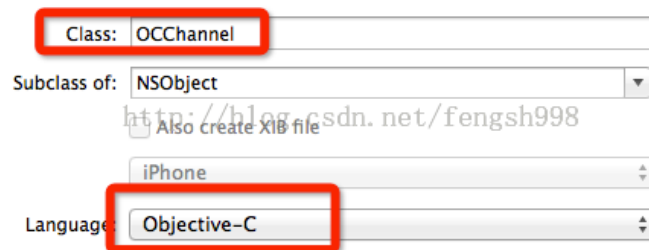
其实, swift也是需要使用头文件进行访问的, 只不过不再需要使用显式的方式使用import进行导入。有两种方式

来实现这个头文件的生成。

方式一：在一个全新的Swift，利用第一次新建提示的方式自动添加桥接头文件。

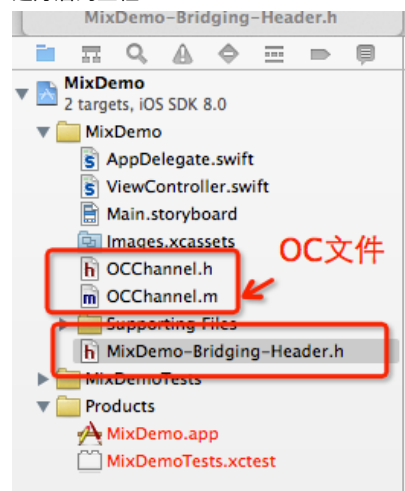


se options for your new file:

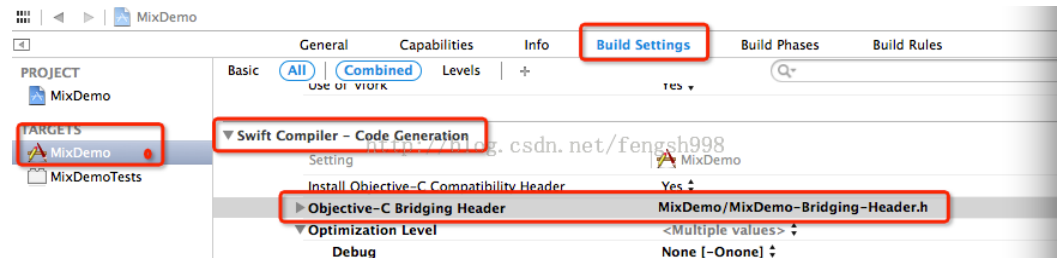


点确定后就会生成一个以<produceName-Bridging-Header.h>的头文件。

建好后的工程：



这里有一个地方需要注意的就是在targets->build settings ->Object-C Bridging Header 设为哪个桥接的头文件即可。



经过上述步骤，桥接文件弄好了就可以

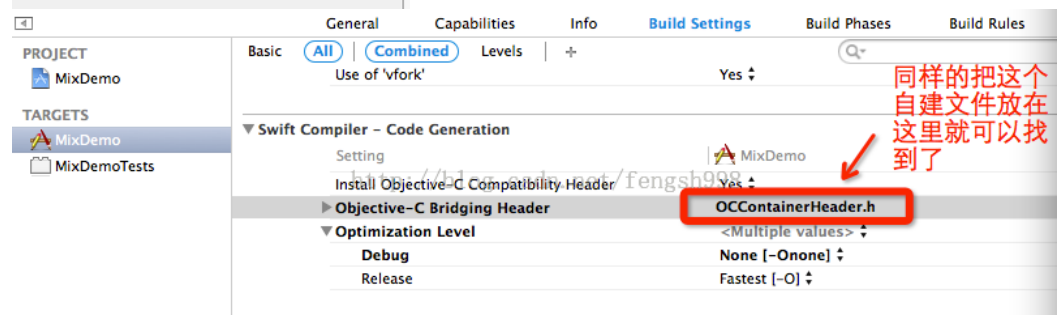
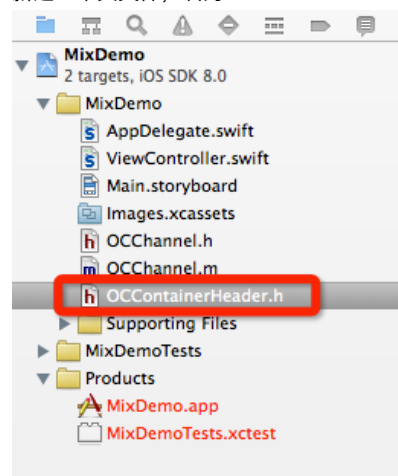
尽情的把想要在swift类中调用的OC头文件放使用import "" 写到这个桥接文件中吧。就像：

```
//  
// Use this file to import your target's public headers that you would like to  
// expose to Swift.  
//MixDemo/MixDemo-Bridging-Header.h  
  
#import "OCChannel.h"
```

同样的，当你知道这个swift搜索头文件的关系后，就不需要再理会这个-Bridging-Header.h的文件了。完全可以手工建一个并取自己喜欢的名字。如：

方式二：

新建一个头文件，名为:OCContainerHeader.h



好了，以上的设置后就完全满足了Swift使用OC写的类了。

```
import UIKit  
  
class ViewController: UIViewController {
```

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    //调用oc类
    var channel = OCChannel()
    println(channel.ChannelChange(10))
    println(channel.ChannelChange(2))
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
    // Dispose of any resources that can be recreated.
}
}
```

好了下面再来看一下**OC如何调用Swift写的类**。（事实上，如果你是一比一抄我这个DEMO，那么恭喜你，在以上你将编译不通过。因为我的OC类中引用了swift写的类，所以你要想运行，就必须把哪个Act的类注释了才行。）

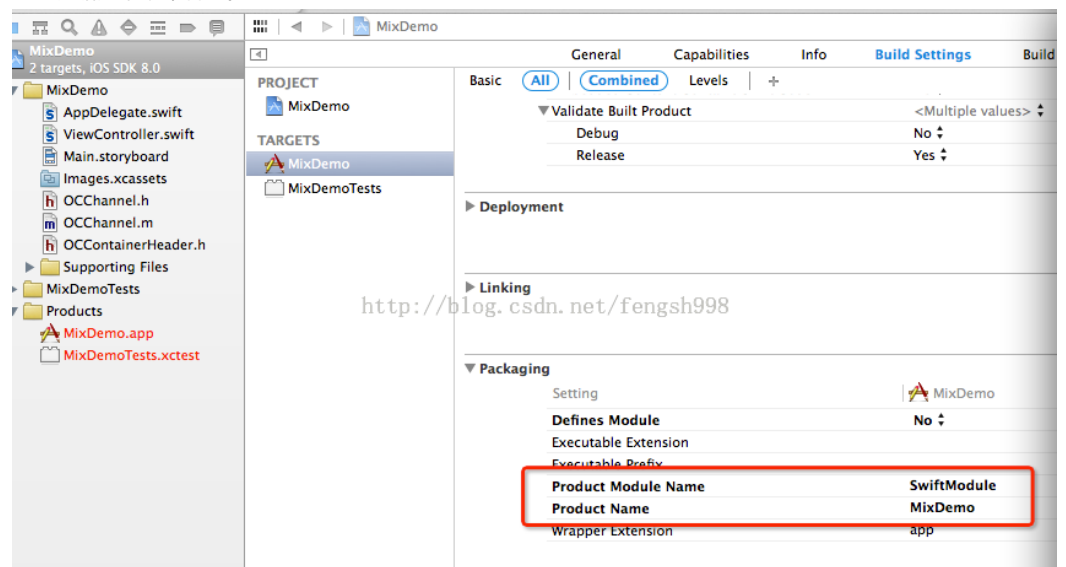
#### OC如何调用Swift写的类

OC要想使用，必须有头文件。而swift文件却没有头文件，所以咱们想必也需要产生一个头文件。但对于OC调用swift的头文件比较特殊。因头文件里面的机制是自动生成的，在不熟悉的，不建议手写。

哪如何产生这个头文件。（注意，系统设置的头文件，在工程中是看不到的。）

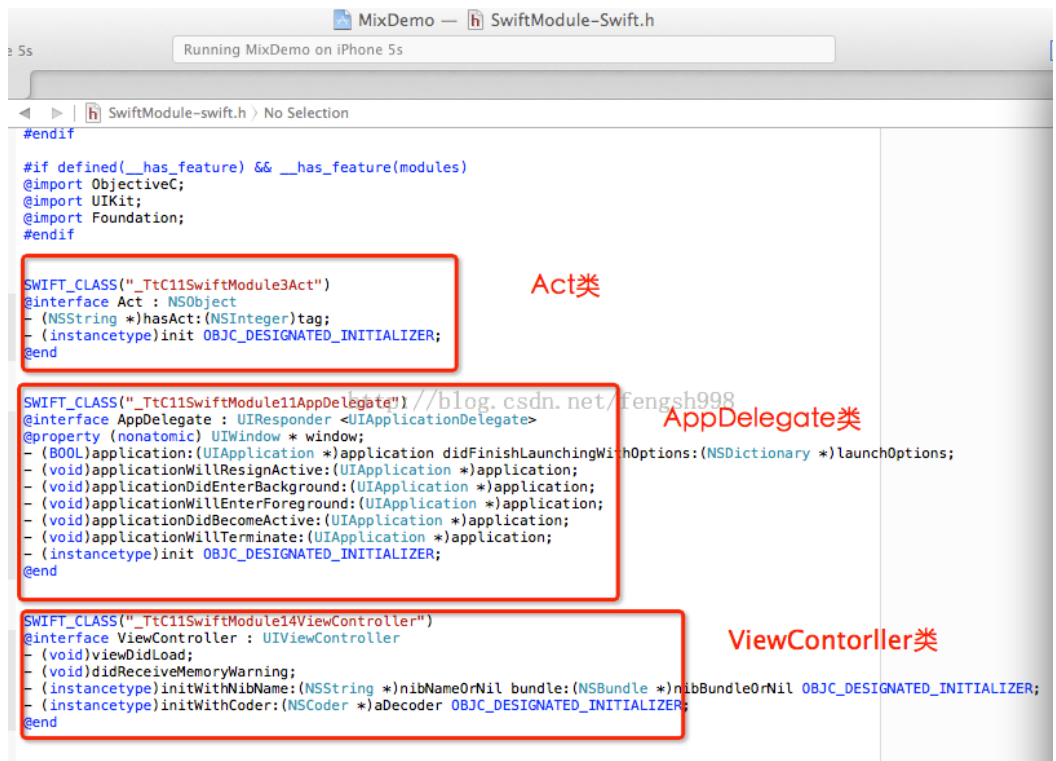
产生步骤：

选中targets->build settings ->packing->Product Module Name 中设置模块名，这个名称很重要 swift的头文件就是根据这个来命名的。





虽然你看图中有这个import "SwiftModule-swift.h"但你在整个工程中是找不到这个文件的，但可以使用CMD+ 鼠标点击可看这个头文件中的内容。



这样，工程中如查Swift要使用OC,则把需要使用的OC类的头文件，全写在MixDemo-Bridging-Header.h里。同样如果OC中所使用的swift类，只需要Clean一把，再编就可以了，但不要忘了导入SwiftModule-swift.h哦（名称自取，但-swift.h是固定的），另外还有一个需要读者注意的。

**注：**

凡是用Swift写的类，如果不继承自NSObject或NSObject的派生类，那么编译后将不会生成对应的转换类。从而使得OC中找不到相应的声明。

如我的例子中 class Act 这样不会被编译到SwiftModule-swift.h中，但写为 class Act : NSObject，就可以编译出相应的声明。另外可以使用@objc加以声明，但这个还是一样，类最好继承NSObject下来。就像下面：

```
import Foundation

@objc(Act)

class Act
{
    func hasAct(tag:Int) -> String
    {
        switch (tag)
        {
            case 1: return "Movie"
        }
    }
}
```

```
        case 2: return "CCTV"
        case 3: return "Sport TV"
        default: return "Area TV"
    }
}

@objc(init)//原本以为加上这个alloc就可以找到,但不行的。。。
init()
{
    println("act constructor is called.")
}

deinit
{
    println("act destroyed is called.")
}
}
```

但是在使用时你就会发现

```
act = [[Act alloc]init]; //报错,找不到alloc, 因此建议大家还是继承NSObject.
```

好吧,今天就写到这里,再研究一下framework之间的混编。。。。。

顶 踩  
8 1

上一篇 初探swift语言的学习笔记八(保留了许多O C的实现)

下一篇 Swift实战

## 猜你在找

微信公众平台开发入门

iOS8-Swift开发教程

微信公众平台深度开发Java版 v2.0 (第一季) 完整版

Swift视频教程

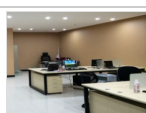
韦东山嵌入式Linux第一期视频



app开发报价



双证在职研究



办公室出租



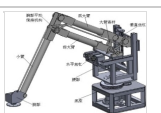
小米笔记本



服务机器人



web开发平台



码垛机器人



[查看评论](#)

\* 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

[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](mailto: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 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 