



iOS SDK 开发者使用文档

Cobub Razor

版本 1.0

| | | | |
|----------------------------------|------|---------|-----|
| 版权所有 © 南京西桥科技有限公司 | | | |
| 作者: | | 主管: | |
| 部门: 项目部 | 签名: | 部门: 项目部 | 签名: |
| 姓名: | | 姓名: | |
| 电话: | | 电话: | |
| 文件: Cobub Razor 开发者使用文档(iOS).doc | 状态: | | |
| 日期: 2012-03-07 | 文件编号 | | |

文档管理

更改记录

| 版本 | 状态 | 日期 | 负责人 | 更改原因 |
|-----|----|------------|------------|------|
| 1.0 | 发布 | 2012-03-07 | 南京西桥科技有限公司 | 发布版本 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

目录

1 概要 4

1.1 文档目的 4

1.2 适用范围 4

1.3 术语和缩写..... 4

1.4 相关文档 4

2 使用说明 4

2.1 建立 APP 4

2.2 使用 SDK 4

2.2.1 下载 SDK..... 4

2.2.2 导入 SDK..... 4

2.2.3 添加依赖框架..... 5

2.2.4 设置 Other Linker Flags..... 5

2.2.5 嵌入代码..... 6

2.2.6 使用自定义事件 6

2.2.7 自动更新..... 7

2.2.8 使用在线参数配置 7

2.2.9 动态调整发送策略 7

1 概要

1.1 文档目的

本文档目的在于为 iOS SDK 开发者提供使用指导。

1.2 适用范围

本文档是适用于 iOS SDK 的使用者。

1.3 术语和缩写

Cobub Razor 移动应用统计分析

1.4 相关文档

无

2 使用说明

2.1 建立 APP

注册并登录 Mobile Applications Analytics 管理平台，在管理平台中选择发布渠道，并完成添加一个新应用，应用添加完成后将可以获得一个应用程序 Key。

2.2 使用 SDK

2.2.1 下载 SDK

下载 SDK 压缩包并将其解压缩。解压缩之后，可以看到两个文件，UMSAgent.h 和 libUMSAgent.a

2.2.2 导入 SDK

在项目工程目录中，选择 Add->Existing files.. 选择上面解压缩后的两个文件。或者直接将两个文件拖拽至目录结构中，选择 Copy Items into destination group's folder(if needed).

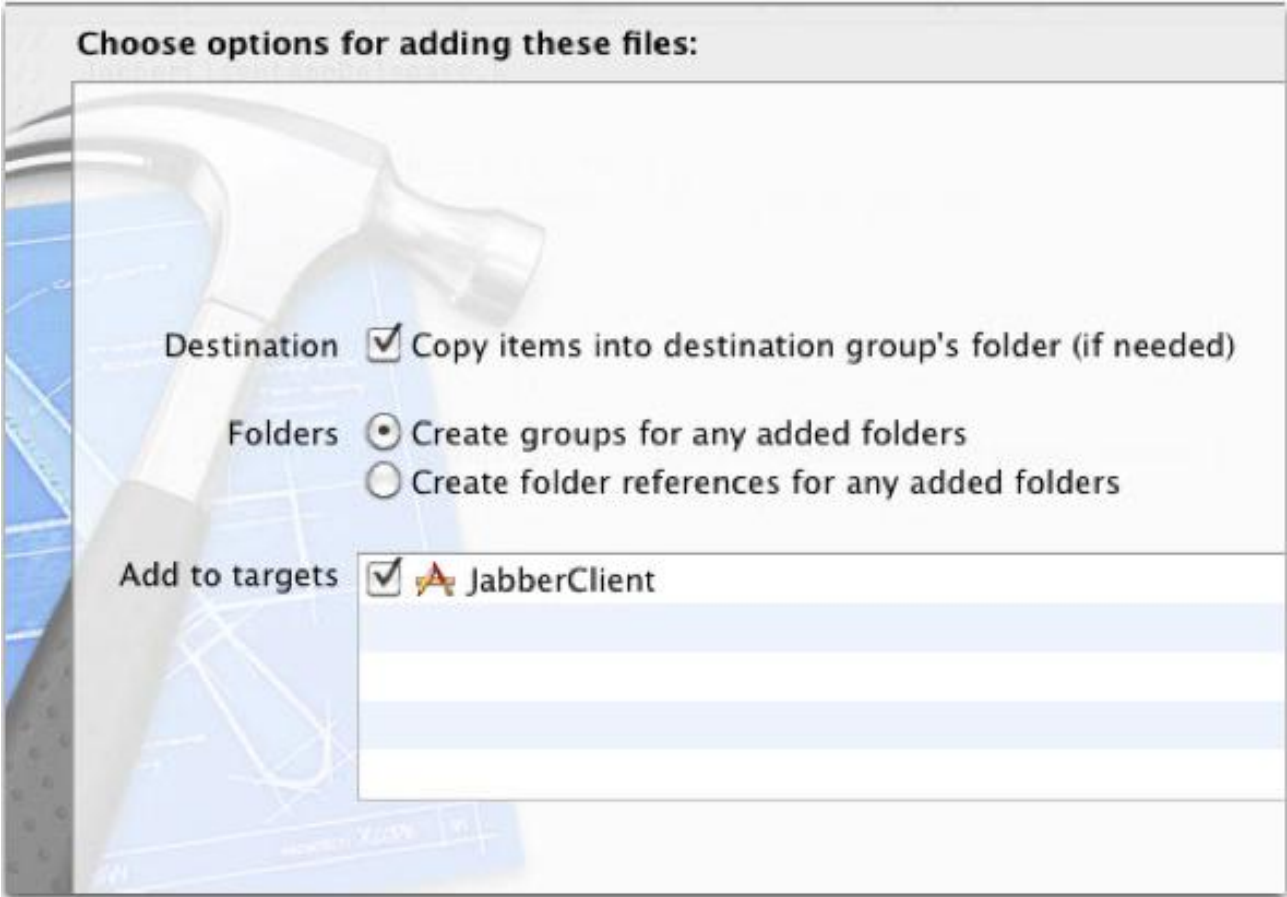


图 2-1 导入 SDK

2.2.3 添加依赖框架

为了获取网络信息,获取运营商标识,您需要添加 CoreTelephony 类库。(见图 2-2 添加 Framework)

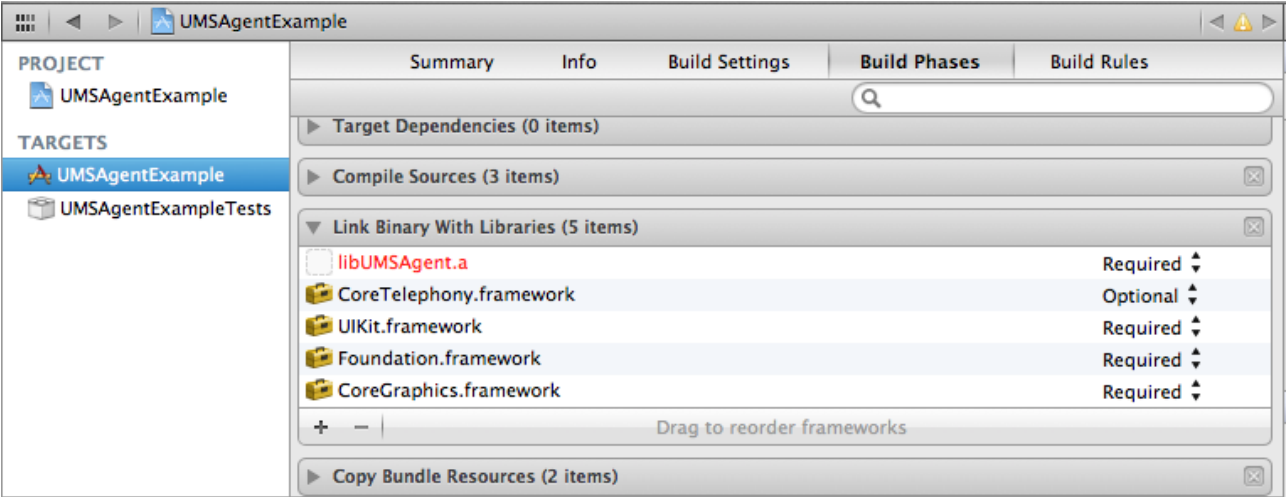


图 2-2 添加 Framework

2.2.4 设置 Other Linker Flags

添加 -all_load 和-ObjC 标记 (见图 2-3 添加 Linker Flags)

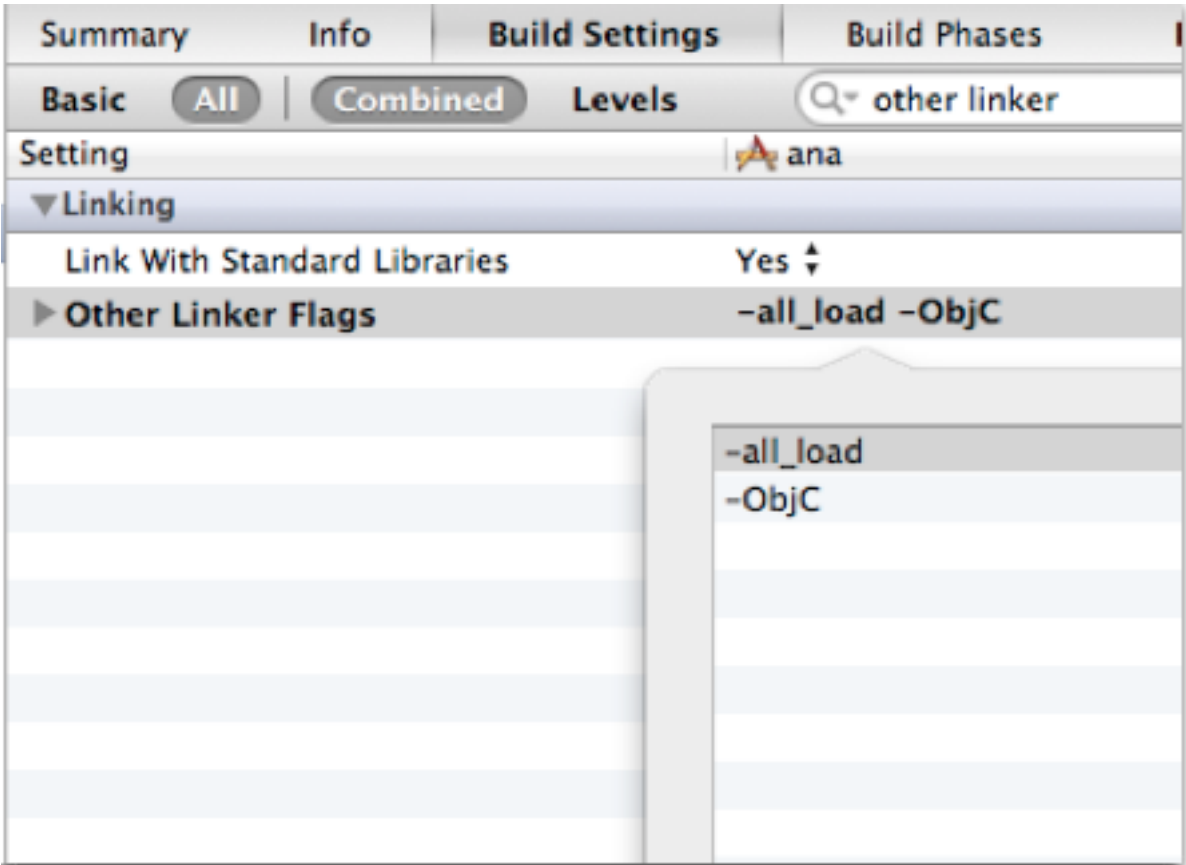


图 2-3 添加 Linker Flags

2.2.5 嵌入代码

在 AppDelegate.h 中添加，#import “UMSAgent.h”。在 AppDelegate.m 中的 didFinishLaunchingWithOptions 函数添加如下函数调用：

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [UMSAgent startWithAppKey:@"UMSAgentKey" ReportPolicy:BATCH];
}
```

图 2-4 设置 UMSAgentKey

其中 UMSAgentKey 是从 Cobub Razor 网站注册应用时获取到的 AgentKey。

其中 BATCH 模式为系统下次启动时一次性上传数据。
如果希望改为实时上传模式，请将 BATCH 改为 REALTIME
建议采用 BATCH 模式，减少 APP 与网络交互，节约流量，增强用户体验。

使用时长统计

Cobub Razor 平台将每次 APP 进入前台和进入后台之间的间隔作为一次生命周期过程。Cobub Razor 会将每次生命周期作为数据提交到 Cobub Razor 后台，为开发者提供相应的统计报告。

2.2.6 使用自定义事件

自定义的事件用来为开发者提供更为详细的用户行为分析。使用前请在 Cobub Razor 后台提前添加指定的事件，并获得事件的 Event_idenfier 后，后台才能进行相应的处理工作，否则后台对客户端未注册事件不进行处理。

2.2.6.1 简单事件

简单事件用来统计指定 event_id 被调用的次数，并生成相应的报告。

```
+(void)postEvent:(NSString *)event_id;
```

2.2.6.2 多标签事件

多标签事件同简单事件，区别在于用户可以为 event_id 设置标签

```
+(void)postEvent:(NSString *)event_id label:(NSString *)label;
```

2.2.6.3 累计事件

累计事件用于多次频繁调用的事件，客户端可以自定义一个计数器，并一次性将结果上传，提高网络效率。

```
+(void)postEvent:(NSString *)event_id acc:(NSInteger)acc;
```

2.2.6.4 累计多标签事件

累计多标签事件综合了多标签事件和累计事件的特性。

```
+(void)postEvent:(NSString *)event_id label:(NSString *)label acc:(NSInteger)acc;
```

2.2.7 自动更新

在 Cobub Razor 后台将版本上传到不同的渠道中，注意后台为每个应用每个渠道提供了一个单独的 Key，在为不同的渠道提供更新时，请注意更换为不同的 Key。

用户使用自动更新功能，需要在调用如下代码：

```
[UMSAgent checkUpdate];
```

注意：此语句需要在执行 SetDelegate 代码之后：

```
[UMSAgent setDelegate:self reportPolicy:REALTIME];
```

2.2.8 使用在线参数配置

用户在服务器端设置了参数值后，客户端可以将在线的参数数据更新到本地使用。使用在线参数需要在 SetDelegate 之后调用如下函数：

```
[UMSAgent setDelegate:self reportPolicy:REALTIME];|  
[UMSAgent setOnLineConfig:false];
```

2.2.9 动态调整发送策略

用户可以通过如下代码的调用来使用服务器端配置的策略来进行数据发送。

同样此功能的使用需要在 SetDelegate 之后立即调用如下代码：

```
[UMSAgent setDelegate:self reportPolicy:REALTIME];|  
[UMSAgent setOnLineConfig:false];
```

2.2.10 记录页面访问

用户可以分别在进入页面和跳出页面时调用 sdk 提供的方法来记录用户页面访问的次数和时间。

```
+(void)startTracPage:(NSString*)page_name;  
+(void)endTracPage:(NSString*)page_name;
```

在进入页面时调用：

```
[UMSAgent endTracPage:@"GWTEST"];
```

在跳出页面时候调用：

```
[UMSAgent startTracPage:@"GWTEST"];
```