

Bugly iOS SDK 高级功能使用指南

功能控制

注意：所有功能控制接口请在初始化 **Bugly** 前调用

- 日志输出控制

开启/关闭 Bugly日志输出,默认关闭。建议 release 时关闭

```
- (void)enableLog:(BOOL)enabled;
```

- 卡顿监测功能控制

开启/关闭 卡顿监测及上报功能，捕获功能默认开启，上报默认开启

```
- (void)enableBlockMonitor:(BOOL)monitor autoReport:(BOOL)reporter;
```

自定义参数

- 设置渠道标识

自定义渠道标识，默认为空值

```
- (void)setChannel:(NSString *)channel;
```

- 设置设备标识

SDK默认使用CFUDID标识设备，注意：平台依据 **deviceId** 统计用户数，如果设置修改，请保证其唯一性

```
- (void)setDeviceId:(NSString *)deviceId;
```

- 设置应用版本

SDK默认读取Info.plist文件中的版本信息,并组装

成 `CFBundleShortVersionString(CFBundleVersion)` 格式

```
- (void)setBundleVer:(NSString *)bundleVer;
```

- 设置用户标识

SDK默认值为 10000

```
- (void)setUserId:(NSString *)userid;
```

- 自定义应用 Bundle Id

```
- (void)setBundleId:(NSString *)bundleId;
```

- 自定义附件

崩溃发生时, 添加附件内容。附件格式为字符串, 字符最大长度为10 * 1024 在回调方法中调用

```
- (void)setAttachLog:(NSString *)attachment;
```

- 自定义 Key Value

```
- (void)setUserData:(NSString *)key value:(NSString *)value;
```

异常回调

在异常发生后进行调用, 应用可以设置回调函数, 并在回调中保存异常现场信息等信息, 异常现场等信息可以通过 `getCrashXXX` 接口获取

使用示例:

```
static int exception_callback_handler() {
    NSLog(@"Crash occur in the app");
    [... setAttachLog:@""];
    return 1;
}
```

```
exp_call_back_func=&exception_callback_handler;
```

配置符号表

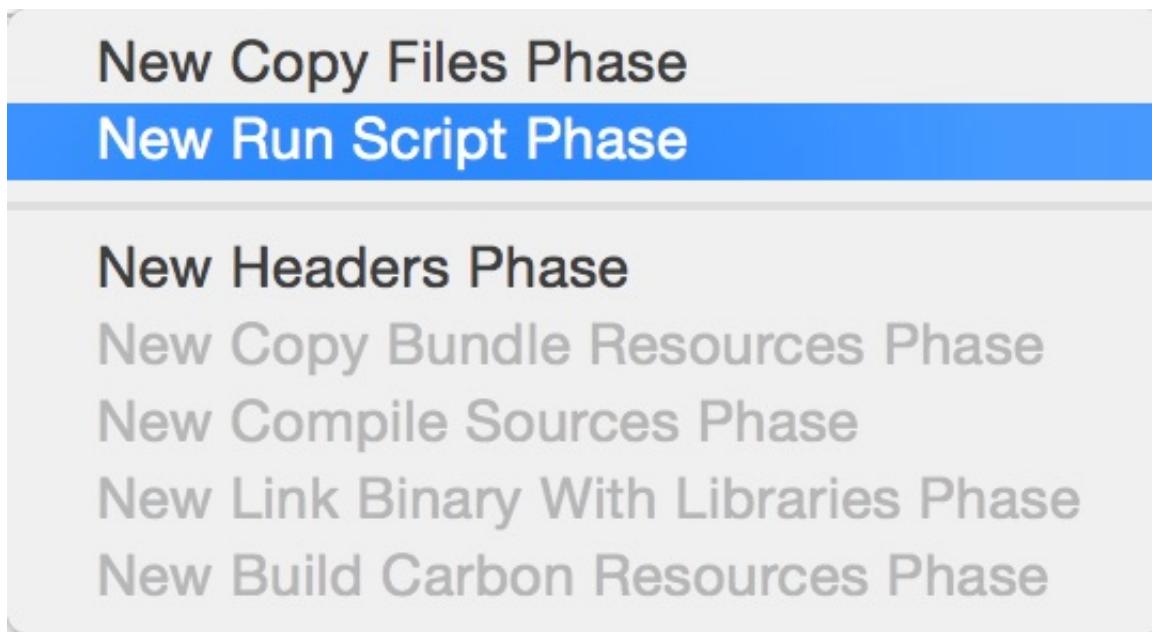
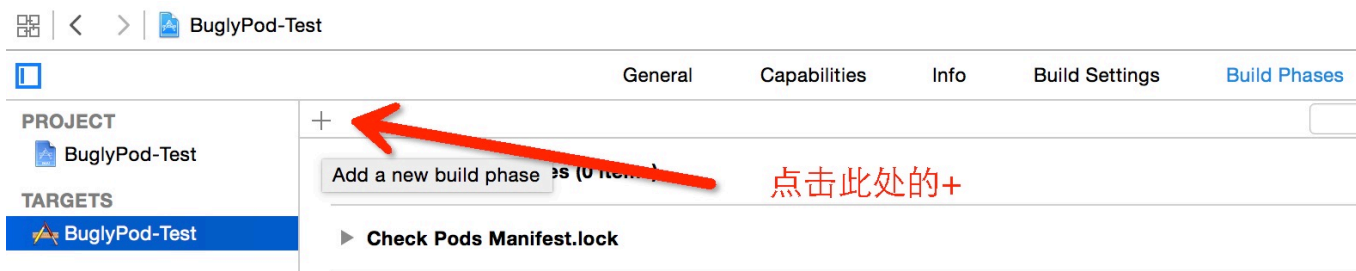
Bugly 提供了两种配置符号表的方式

- sh脚本自动上传
- 手动上传

sh脚本自动上传

- 我们的符号表提取工具依赖 java 运行环境 [java 环境下载](#)
- 下载解压 [自动配置符号表 zip 文件](#)
- 把符号表提取工具 `buglySymbolIOS.jar` 保存在 `~/bin` 目录下 (~为用户主目录, 如用户主目录没有 `bin` 文件夹,请自行创建)

- 在 Xcode 工程对应 Target 的 Build Phases 中新增 Run Script Phase



- 打开 `dsym_upload.sh`，复制所有内容，在新增的 Run Script Phase 中粘贴
- 修改新增的 Run Script 中的 `<YOUR_APP_ID>` 为您的 appid, `<YOUR_APP_KEY>` 为您的 appkey, `<YOUR_BUNDLE_ID>` 为 app 的 Bundle Id

▼ Run Script

Shell `/bin/sh`

```

1 #!/bin/sh
2 # Copyright 2015 Tencent.com
3 # http://bugly.qq.com
4 #
5 #请替换为您的APPID, APPKEY 及 app 的 bundle id
6
7 APP_ID="<YOUR APP ID HERE>"
8 API_KEY="<YOUR APP KEY HERE>"
9 BUNDLE_ID="<YOUR BUNDLE ID HERE>"
10 # 模拟器编译是否上传, 1=上传 0=不上传

```

脚本默认在 **Debug** 模式及 **模拟器编译** 情况下不会上传符号表，在需要上传的时候，请修改下列选项

- Debug模式编译是否上传，1 = 上传 0 = 不上传，默认不上传

`UPLOAD_DEBUG_SYMBOLS=0`

- 模拟器编译是否上传，1 = 上传 0 = 不上传，默认不上传

`UPLOAD_SIMULATOR_SYMBOLS=0`

至此，自动上传符号表脚本配置完毕，Bugly 会在每次 Xcode 工程编译后自动完成符号表配置工作。

如果不想自动上传模拟器编译后的符号文件，请修改脚本 `UPLOAD_SIMULATOR_SYMBOLS=0`

手动上传

关于手动上传符号表，请参阅 [手动符号表配置指南](#)

更多功能请查阅 Bugly 头文件 `CrashReporter.h`