

JCSDK Docking Documentation

Description

Version: 1.0.0

目录

I 、 SDK Introduction :	2
1.1、 Supported ad types:	2
1.2、 version:	2
II 、 SDK Access Configuration : (Contains bridging and configuration cs)	3
2.1、 info.plist configuration:	3
2.2、 build setting configuration:	4
2.3、 Importing related SDKs:	4
2.4、 Required system support library:	4
2.5、 JCIOSConfig.plist file description:	5
III 、 unity access configuration and Api description :	6
<u>3.1 JC_unityAdApi.h Interface Description:</u>	<u>6</u>
3.2 JC_unityCallBackApi.h Interface Description:	8
IV 、 Advertising interface API and callback usage	10
4.1、 Initialization and splash ad api descriptions:	10
4.2、 banner api description:	11

4.3、intersitial api description:	12
4.4、rewardVideo api description:	13
4.5、Callback Example:.....	14
V 、Related Error Reporting	16
<u>VI 、iOS14 support</u>	20

I 、SDK Introduction :

JCSDK is a set of advertising SDK provided by MS Company, which integrates the advertising SDKs of major advertisers and related statistical SDKs to facilitate the joint operation and data analysis of in-app advertising between platforms.

1.1、Supported ad types:

splash Ads, banner ads, rewardVideo ads, interstitial ads, native ads

1.2、version:

See: [version](#)

II 、 SDK Access Configuration : (Contains bridging and configuration cs)

We provide bridge and configuration files, please refer to [cs file](#) and download:



2.1、 info.plist configuration:

Support http network configuration

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
```

Google-related parameter configuration:

```
<key>GADApplicationIdentifier</key>
<string>ca-app-pub-9488501426181082/7319780494</string>
<key>GADIsAdManagerApp</key>
```

<true/>

Get location permission configuration

<key>NSLocationWhenInUseUsageDescription</key>

<string>The app needs to get your location</string>

Get IDFA permissions , iOS14support

<key>NSUserTrackingUsageDescription</key>

<string>This identifier will be used to deliver personalized ads to you.</string>

2.2、 build setting configuration:

Xcode - build setting :

Bitcode set NO,

“Other linker flags” add “-ObjC”

2.3、 Importing related SDKs:

Here is the MS ad-supported library and documentation: check out the

[JCSDK](#) download

JCSDK.framework、JCiOSConfig.plist























Third-party advertising support library and related documents:

We only provide manual import for now, check out the [ADThirdParty_SDK](#) file and download.

Data platform library and related documents : check out the

[DataCollection_SDK](#) file download.

2.4、 Required system support library:

▼ Frameworks, Libraries, and Embedded Content	
Name	Embed
 Accelerate.framework	Do Not Embed ⇅
 AdSupport.framework	Do Not Embed ⇅
 AVFoundation.framework	Do Not Embed ⇅
 CoreGraphics.framework	Do Not Embed ⇅
 CoreLocation.framework	Do Not Embed ⇅
 CoreMedia.framework	Do Not Embed ⇅
 CoreMotion.framework	Do Not Embed ⇅
 CoreTelephony.framework	Do Not Embed ⇅
 iAd.framework	Do Not Embed ⇅
 libbz2.tbd	
 libc++.tbd	
 libresolv.9.tbd	
 libsqlite3.tbd	
 libxml2.tbd	
 libz.tbd	
 MessageUI.framework	Do Not Embed ⇅
 SafariServices.framework	Do Not Embed ⇅
 Security.framework	Do Not Embed ⇅
 SystemConfiguration.framework	Do Not Embed ⇅
 UIKit.framework	Do Not Embed ⇅
 VideoToolbox.framework	Do Not Embed ⇅
 WebKit.framework	Do Not Embed ⇅
+ —	

AppTrackingTransparency.framework (iOS14 support)

2.5、 JCIOSConfig.plist file description:

V1.0.0 add

Item	Value
appid	Appid required for JCSDK initialization
channelid	ChannelId required for JCSDK initialization
ReYunAppID	Appid required for reyun initialization
ReYunChannelID	channelId required for reyun initialization
UmengAppID	Appid required for UMeng initialization
ShuShuAppID	Appid required for 数数 initialization
TalkingDataAppID	Appid required for TalkingData initialization

V2.0.0 add

Item	Value
KochavaAppID	Appid required for Kochava initialization
TenJinAppID	Appid required for tenjin initialization
ShowSplashFirst	Whether to display an open-screen ad when opening the app for the first time, bool type: YES/NO
LogLevel	Log level: string type. 1. Close. 2. Open JC log. 3. Open JC+ad log. 4. Open JC+ad+data log

III 、 unity access configuration and Api description :

3.1 JC_unityAdApi.h Interface Description:

V1.0.0

```
-(void)initJCSDKWithLog:(BOOL)isOpenLog  
isFirstShowSplash:(BOOL)isShow  
splashClose:(unityBlock)block;
```

V2.0.0 change init

```
-(void)initJCSDKWithUnityShow:(unityBlock)block;
```

```
/// Intersitial Ads isReady
```

```
bool isReadyIntersitial();
```

```
/// show Intersitial Ads
```

```
void showIntersitial();
```

```
/// rewardVideo Ads isReady
```

```
bool isReadyRewardVideo();
```

```
/// show rewardVideo Ads
```

```
void showRewardVideo();
```

```
/// isReady - banner
```

```
bool isReadyBanner();
```

```
/// show banner Ads
```

```
void showBannerView();
```



```

/// remove banner Ads
void removeBannerView();

/// Send Event UMeng、 talkingData
/// @param event event
/// @param jsonEventInfo key-value converted json string, if there is no
content to pass, you can set a null value
void sendEvent(char *event, char *jsonEventInfo);

```

3.2 JC_unityCallbackApi.h Interface Description:

```

/// Sign up for a callback monitor to be invoked before creating a bridge
back to the advertiser.
void RegistCallbacknotifition();

/// splash callback bridge
/// @param failLoad
/// @param didShow
/// @param didClick
/// @param didClose
void splash_CallBack(ResultHandler failLoad, ResultHandler
didShow, ResultHandler didClick, ResultHandler didClose);

/// intersitial callback bridge
/// @param failLoad
/// @param didShow
/// @param failToShow
/// @param didClose

```

```
/// @param didClick
/// @param failToPlayVideo
/// @param startPlayingVideo
/// @param endPlayingVideo
void Intersitial_Callback(ResultHandler
failLoad,ResultHandler didShow, ResultHandler failToShow,
ResultHandler didClose,ResultHandler
didClick,ResultHandler failToPlayVideo, ResultHandler
startPlayingVideo, ResultHandler endPlayingVideo);
```

```
/// banner callback bridge
/// @param failLoad load
/// @param didShow
/// @param didClick
/// @param didAutoRefresh
/// @param tapCloseBtn
/// @param failToAutoRefresh
void banner_Callback(ResultHandler failLoad,ResultHandler
didShow,ResultHandler didClick,ResultHandler
didAutoRefresh, ResultHandler tapCloseBtn, ResultHandler
failToAutoRefresh);
```

```
/// rewardVideo callback bridge
/// @param failLoad
/// @param didRewardSuccess
/// @param didClose
/// @param didClick
/// @param failToPlayVideo
/// @param startPlayingVideo
/// @param endPlayingVideo
```

```

void rewardVideo_CallBack(ResultHandler
failLoad,ResultHandler didRewardSuccess, ResultHandler
didClose,ResultHandler didClick,ResultHandler
failToPlayVideo, ResultHandler startPlayingVideo,
ResultHandler endPlayingVideo);

/// native  callback bridge (Not in use yet)
/// @param failLoad
/// @param didShow
/// @param didClick
/// @param startPlayingVideo
/// @param endPlayingVideo
/// @param tapCloseBtn
/// @param enterFullScreenV
/// @param exitFullScreenV
void native_CallBack(ResultHandler failLoad,ResultHandler
didShow, ResultHandler didClick, ResultHandler
startPlayingVideo, ResultHandler
endPlayingVideo,ResultHandler tapCloseBtn,ResultHandler
enterFullScreenV,ResultHandler exitFullScreenV);

```

IV 、 Advertising interface API and callback usage

4.1、 Initialization and splash ad api descriptions:

header file:

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

We internally demonstrate splash at program startup and when the program returns to the foreground.

4.1.1 、 If your app is a game app, please implement it in UnityAppController.mm. In order to display the splash ad in front of unity, you need to find the following code and replace it, either insert it by force using unity code, or convert it to xcode project and write it manually.

```
330         _startUnityScheduled = true;
331         //[self performSelector: @selector(startUnity:) withObject: application
332             afterDelay: 0];
333         [self performSelector: @selector(initSDKWithApplication:) withObject:
334             application afterDelay: 0];
335     }
336     _didResignActive = false;
337 }
338 -(void)initSDKWithApplication:(UIApplication*)application{
339     [[JC_unityAdApi getInstance]initJCSdkWithLog:YES isFirstShowSplash:NO
340         splashClose:^(BOOL isOk) {
341         [self performSelector: @selector(startUnity:) withObject: application
342             afterDelay: 0];
343     }];
344 }
```



4.2、 banner api description:

Showbanner Recommended Call Order: isReady (YES) - show , When you use "removebanner", the ad caching logic is automatically called internally, so don't call the banner's other api (which has a data buffer) right after you use "removebanner"!

```

[DllImport("__Internal")]
static extern void showBannerView();

[DllImport("__Internal")]
static extern void removeBannerView();

public static void ShowBanner()
{
    showBannerView();
}

public static void removeBanner()
{
    removeBannerView();
}

```

4.3、intersitial api description:

It is advisable to determine whether IsInterReady has internal advertising space available, and then show the advertisement.

```

[DllImport("__Internal")]
static extern bool isReadyIntersitial();

[DllImport("__Internal")]
static extern void showIntersitial();

```

```

public static bool IsInterReady()
{
    var value = isReadyIntersitial();
    Debug.Log("-----> IsInterReady:" + value);
    return value;
}

public static void ShowInter()
{
    showIntersitial();
}

```

4.4、rewardVideo api description :

It is advisable to determine whether IsRewardVReady has internal advertising space available, and then show the advertisement.

```

[DllImport("__Internal")]
static extern bool isReadyRewardVideo();

[DllImport("__Internal")]
static extern void showRewardVideo();

public static bool IsRewardVReady()
{
    var value = isReadyRewardVideo();
    Debug.Log("-----> isReadyRewardV:" + value);
    return value;
}

```

```
public static void ShowRewardV()
{
    showRewardVideo();
}
```

4.5、 Callback Example:

intersitial callback example:

```
[DllImport("__Internal")]
static extern void Intersitial_CallBack(IntPtr failLoad, IntPtr didShow,
IntPtr failToShow, IntPtr didClose, IntPtr didClick, IntPtr failToPlayVideo,
IntPtr startPlayingVideo, IntPtr endPlayingVideo);
```

```
//Register
var handler11 = new ResultHandler(interFailLoad);
var fp11 = Marshal.GetFunctionPointerForDelegate(handler11);
var handler12 = new ResultHandler(interDidShow);
var fp12 = Marshal.GetFunctionPointerForDelegate(handler12);
var handler13 = new ResultHandler(interFailToShow);
var fp13 = Marshal.GetFunctionPointerForDelegate(handler13);
var handler14 = new ResultHandler(interDidClose);
var fp14 = Marshal.GetFunctionPointerForDelegate(handler14);
var handler15 = new ResultHandler(interDidClick);
var fp15 = Marshal.GetFunctionPointerForDelegate(handler15);
var handler16 = new ResultHandler(interFailToPlayVideo);
var fp16 = Marshal.GetFunctionPointerForDelegate(handler16);
var handler17 = new ResultHandler(interStartPlayingVideo);
```

```
var fp17 = Marshal.GetFunctionPointerForDelegate(handler17);  
var handler18 = new ResultHandler(interEndPlayingVideo);  
var fp18 = Marshal.GetFunctionPointerForDelegate(handler18);  
Intersitial_CallBack(fp11, fp12, fp13, fp14, fp15, fp16, fp17, fp18);
```

```
//  
[MonoPInvokeCallback(typeof(ResultHandler))]  
static void interEndPlayingVideo(string resultString)  
{  
    Debug.Log("intersitial callback----->interEndPlayingVideo");  
}  
[MonoPInvokeCallback(typeof(ResultHandler))]  
static void interStartPlayingVideo(string resultString)  
{  
    Debug.Log("intersitial callback----->interStartPlayingVideo");  
}  
[MonoPInvokeCallback(typeof(ResultHandler))]  
static void interFailToPlayVideo(string resultString)  
{  
    Debug.Log("intersitial callback----->interFailToPlayVideo");  
}  
[MonoPInvokeCallback(typeof(ResultHandler))]  
static void interDidClick(string resultString)  
{  
    Debug.Log("intersitial callback----->interDidClick");  
}  
[MonoPInvokeCallback(typeof(ResultHandler))]  
static void interDidClose(string resultString)  
{
```



```

        Debug.Log("intersitial callback----->interDidClose");
    }
    [MonoPInvokeCallback(typeof(ResultHandler))]
    static void interFailToShow(string resultString)
    {
        Debug.Log("intersitial callback----->interFailToShow");
    }

    [MonoPInvokeCallback(typeof(ResultHandler))]
    static void interDidShow(string resultString)
    {
        Debug.Log("intersitial callback----->interDidShow");
    }

    [MonoPInvokeCallback(typeof(ResultHandler))]
    static void interFailLoad(string resultString)
    {
        Debug.Log("intersitial callback----->interFailLoad");
    }
}

```

V 、 Related Error Reporting

5.1、 Crash on startup

The application crashes at startup because you are missing certain configurations, we give a few examples here

Missing "-ObjC flag" configuration item for "Other Linker Flags" in "Build Settings".

Solutions: Add "-ObjC"

The program starts and crashes after introducing the "Admob" SDK.

Solution: Add the key Google needs to "info.plist".

```
<key>GADApplicationIdentifier</key>
```

```
<string>ca-app-pub-9488501426181082/7319780494</string>
```

```
<key>GADIsAdManagerApp</key> <true/>
```

Introduction of "快手/KS" SDK compilation crash

Solution: Modify "Embed" in "KSAdSDK.framework" to "Embed&Sign".

Crashes while displaying a wide-spread incentive video or interstitial ad
-[AppDelegate window]

Solution: Add "window" attribute to "AppDelegate.h".

4.2 AppPacket failure/packet submission failure

KSadSDK contains x86 binaries, the Apple Store does not support emulator resources.

solutions:

Build Phase -> New Run Scrip Phase

After adding "new Run Script Phase", "Run Script" will appear, then add a

script code as follows

```
APP_PATH="${TARGET_BUILD_DIR}/${WRAPPER_NAME}"

# This script loops through the frameworks embedded in the
application and

# removes unused architectures.

find "$APP_PATH" -name '*.framework' -type d | while read
-r FRAMEWORK

do

    FRAMEWORK_EXECUTABLE_NAME=$(defaults read
"$FRAMEWORK/Info.plist" CFBundleExecutable)

    FRAMEWORK_EXECUTABLE_PATH="$FRAMEWORK/$FR
AMEWORK_EXECUTABLE_NAME"

    echo "Executable is
$FRAMEWORK_EXECUTABLE_PATH"

    EXTRACTED_ARCHS=()
```

```
for ARCH in $ARCHS

do

    echo "Extracting $ARCH from
$FRAMEWORK_EXECUTABLE_NAME"

    lipo -extract "$ARCH"
"$FRAMEWORK_EXECUTABLE_PATH" -o
"$FRAMEWORK_EXECUTABLE_PATH-$ARCH"

EXTRACTED_ARCHS+=("$FRAMEWORK_EXECUTABLE_P
ATH-$ARCH")

done


echo "Merging extracted architectures: ${ARCHS}"

lipo -o "$FRAMEWORK_EXECUTABLE_PATH-merged"
-create "${EXTRACTED_ARCHS[@]}"

rm "${EXTRACTED_ARCHS[@]}"
```

```
echo "Replacing original executable with thinned
version"

rm "$FRAMEWORK_EXECUTABLE_PATH"

mv "$FRAMEWORK_EXECUTABLE_PATH-merged"
"$FRAMEWORK_EXECUTABLE_PATH"

done
```

After adding the script block, check the "Run script only when installing" box and repackage it for submission.

VI 、iOS14 support

Please refer to the [JCSDK iOS14](#) support documentation for details.