



媒体云播放器 iOS SDK 用户手册

V1.1

发布日期： 2013年8月8日

百度开发者中心

（版权所有，翻版必究）

修改记录

No	修改后版本号	修改内容简介	修改日期
1	1.0	完成 iOS SDK 用户手册	2013-07-04
2	1.1	新增 shouldAutoClearRender 属性	2013-08-08

目录

第 1 章 简介..... 4

第 2 章 阅读对象..... 5

第 3 章 播放器 SDK 功能说明 5

第 4 章 接口说明..... 6

第 5 章 开发前准备 8

 5.1 运行环境 8

 5.2 申请及权限开通 8

 5.3 文档使用 8

 5.3.1 使用 docset 8

 5.3.2 html 查看 8

第 6 章 使用 SDK 开发应用 9

 6.1 添加依赖 9

 6.1.1 使用真机调试、发布..... 9

 6.1.2 使用模拟器调试、发布..... 10

 6.2 调用 API..... 10

 6.2.1 获取播放地址..... 10

 6.2.2 传入 API key 10

 6.2.3 初始化 CyberPlayerController..... 10

 6.2.4 设置播放地址..... 11

 6.2.5 播放视频..... 12

 6.2.6 暂停播放..... 12

 6.2.7 改变播放位置..... 13

 6.2.8 停止播放..... 13

第 7 章 播放信息统计13

第 8 章 联系我们.....13

第1章 简介

百度媒体云播放器 iOS SDK（以下简称“播放器 SDK”）是百度官方推出的 iOS 平台使用的软件开发工具包（SDK），为 iOS 开发者提供简单、快捷的接口，帮助开发者实现 iOS 平台上的媒体播放应用开发。

播放器 SDK 内嵌百度自主研发的 T5 播放内核，对目前主流的本地和网络媒体都提供了良好的功能支持，弥补了系统播放器在媒体支持格式上的不足，并在兼容性、稳定性和响应速度上有明显的提高。

播放器 SDK 提供了与系统播放器 `MPMoviePlayerController` 高度相似的调用接口，非常方便用户使用。

播放器 SDK 的完整下载包是 `Baidu-T5Player-SDK-iOS-1.1.zip`，解压缩后包含 `demo`、`lib`、`doc`、SDK 用户手册、SDK 版本说明五个部分，解压后的目录结构如下所示：

- `demo`：主要存放 1 个 iOS 示例工程，可以帮助用户了解如何使用该 SDK。`SampleOne` 展示了如何使用 `MPMoviePlayerController` 和 `CyberPlayerController` 来播放视频。
- `lib`：主要存放播放器 SDK 的静态 `.a` 文件，包含两个文件夹，分别为 `Release-iphonesos` 和 `Release-iphonesimulator`，分别为真机和模拟器版本静态库。
- `doc`：该目录存放类参考文档。

第2章 阅读对象

本文档面向所有使用该 SDK 的开发人员、测试人员、合作伙伴以及对此感兴趣的其他用户，要求读者具有一定的 iOS 编程经验。

第3章 播放器 SDK 功能说明

百度媒体云播放器 iOS SDK（以下简称“播放器 iOS SDK”）是百度开放云推出的 iOS 平台视频播放器软件开发工具包（SDK），为 iOS 开发者提供简单、便捷的开发接口，帮助开发者在 iPad/iPhone/iPod 和 Apple TV 设备上实现媒体播放功能。播放器 SDK 内嵌百度自主研发的 T5 播放内核，提供简单、便捷的媒体应用开发能力，具有以下的特色功能：

- 本地全媒体格式支持

突破 iOS 平台对视频格式的限制，支持目前所有主流的媒体格式（mp4、avi、wmv、flv、mkv、mov、rmvb 等）。

- 支持广泛的流式视频格式

支持多种格式文件渐进式和流式播放：HLS、RTMP、HTTP Pseudo-Streaming（伪流）。

- 性能强大

资源 CPU/内存占用率低，视频加载速度快。

- 低门槛、高灵活度实现播放功能

简单易用的 API，提供快速媒体播放功能开发和示例。

第4章 接口说明

与 iOS 系统视频播放器 [MPMoviePlayerController](#) 类对比，CyberPlayerController 有如下异同，具体属性、方法请参考《类参考文档》。

表 1 属性对比

type	MPMoviePlayerController	CBVideoPlayerController
property	contentURL	contentURL
	N/A	contentString
	movieSourceType	N/A
	movieMediaTypes	N/A
	allowsAirPlay	N/A
	airPlayVideoActive	N/A
	naturalSize	naturalSize
	N/A	videoWidth
	N/A	videoHeight
	fullscreen	N/A
	scalingMode	scalingMode
	controlStyle	N/A
	duration	duration
	playableDuration	playableDuration
	view	view
	backgroundView	N/A
	loadState	N/A
	playbackState	playbackState
	initialPlaybackTime	initialPlaybackTime
	endPlaybackTime	N/A
	shouldAutoplay	shouldAutoplay
	N/A	shouldAutoClearRender
	readyForDisplay	N/A
	repeatMode	N/A
	isPreparedToPlay	isPreparedToPlay
	accessLog	N/A
	errorLog	N/A
	currentPlaybackRate	N/A
	currentPlaybackTime	currentPlaybackTime

表 2 方法对比

type	MPMoviePlayerController	CBVideoPlayerController
method	- initWithContentURL:	- initWithContentURL:
	N/A	- initWithContentString:
	- setFullscreen:animated:	N/A
	- timedMetadata	N/A
	- beginSeekingBackward	N/A
	- beginSeekingForward	N/A
	- endSeeking	N/A
	N/A	- seekto:

	- thumbnailImageAtTime:timeOption:	N/A
	- requestThumbnailImagesAtTimes:timeOption:	N/A
	- cancelAllThumbnailImageRequests	N/A
	- play	- play
	N/A	- start
	- pause	- pause
	- stop	- stop
	-prepareToPlay	- prepareToPlay
	N/A	- getSDKVersion
	N/A	+ setBAEAPIKey: SecretKey:

表 3 常量对比

type	MPMoviePlayerController	CBVideoPlayerController
Constants	MPMovieLoadState	N/A
	MPMovieControlStyle	N/A
	MPMovieFinishReason	N/A
	MPMoviePlaybackState	CBPMoviePlaybackState
	MPMovieRepeatMode	N/A
	MPMovieScalingMode	CBPMovieScalingMode
	MPMovieTimeOption	N/A
	MPMovieMediaTypeMask	N/A
	MPMovieSourceType	N/A
	Thumbnail Notification User Info Keys	N/A
	Fullscreen Notification Keys	N/A
	Playback Finished Notification Key	N/A
	MPMovieControlMode	N/A

表 4 通知对比

type	MPMoviePlayerController	CBVideoPlayerController
Notifications	MPMovieDurationAvailableNotification	CyberPlayerLoadDidPreparedNotification
	MPMovieMediaTypesAvailableNotification	N/A
	MPMovieNaturalSizeAvailableNotification	CyberPlayerLoadDidPreparedNotification
	MPMoviePlayerContentPreloadDidFinishNotification	N/A
	MPMoviePlayerDidEnterFullscreenNotification	CyberPlayerLoadDidPreparedNotification
	MPMoviePlayerDidExitFullscreenNotification	N/A
	MPMoviePlayerIsAirPlayVideoActiveDidChangeNotification	N/A
	MPMoviePlayerLoadStateDidChangeNotification	N/A
	MPMoviePlayerNowPlayingMovieDidChangeNotification	N/A
	MPMoviePlayerPlaybackDidFinishNotification	CyberPlayerPlaybackDidFinishNotification
	MPMoviePlayerPlaybackStateDidChangeNotification	CyberPlayerPlaybackStateDidChangeNotification

	MPMoviePlayerScalingModeDidChangeNotification	N/A
	MPMoviePlayerThumbnailImageRequestDidFinishNotification	N/A
	MPMoviePlayerWillEnterFullscreenNotification	N/A
	MPMoviePlayerWillExitFullscreenNotification	N/A
	MPMovieSourceTypeAvailableNotification	N/A
	MPMoviePlayerReadyForDisplayDidChangeNotification	N/A
	N/A	CyberPlayerStartCachingNotification
	N/A	CyberPlayerGotCachePercentNotification
	N/A	CyberPlayerPlaybackErrorNotification
	N/A	CyberPlayerSeekingDidFinishNotification

第5章 开发前准备

5.1 运行环境

本 SDK 可运行于 iOS 4.3 及以上版本。

5.2 申请及权限开通

开发者需要使用百度账号登录[百度开发者中心](#)注册成为百度开发者并创建应用，方可获取应用 ID、对应的 API Key（即：ak）及 Secret Key（即：sk）等信息。具体信息，请参考[百度开发者中心](#)上的“[创建应用](#)”的相关介绍。

SDK 认证时必须传入 ak 及 sk（只需前 16 位）参数。

5.3 文档使用

5.3.1 使用 docset

从 Baidu-T5Player-SDK-iOS-1.1.zip 包中解压缩，在 Baidu-T5Player-SDK-iOS-1.1/doc 目录下有 com.baidu.cyberplayer.docset 文件。将该文件复制到 /Users/abc/Library/Developer/Shared/Documentation/DocSets/com.baidu.cyberplayer.docset，其中 abc 为您的用户名，并确保 /Users/abc/Library/Developer/Shared/Documentation/DocSets/com.baidu.cyberplayer.docset 路径下为 Content 文件夹。这样您就可以在 xcode 文档中心查看 CyberPlayerController 类接口文档。

5.3.2 html 查看

从 Baidu-T5Player-SDK-iOS-1.1.zip 包中解压缩，在 Baidu-T5Player-SDK-iOS-1.1/doc/html 路径下有 index.html，双击打开使用浏览器即可查看 CyberPlayerController 类接口文档。

第6章 使用 SDK 开发应用

首先创建您的应用和 target，由于当前的 CyberPlayerController 只支持 armv7 架构，所以在导入相关静态库前，您需要在 Xcode 中配置您的 target。选择 Build Settings->Architectures->Valid Architectures，将该选项的 Debug/Release 键都置为 armv7（即去掉 armv7s 的支持）。

6.1 添加依赖

播放器 SDK 依赖的框架如表 5 所示。

表 5 SDK 依赖的框架

框架名	含义
OPENAL.framework	音频 API
CoreGraphics.framework	轻量级 2D 渲染
OpenGL.framework	三维图形 API
UIKit.framework	界面 API
QuartzCore.framework	图像视频处理 API
libbz2.1.0.dylib	压缩工具
libz.dylib	
AudioToolbox.framework	音频控制 API
Foundation.framework	基本工具
MobileCoreServices.framework	libASIHTTP.a 需要使用
CFNetwork.framework	
SystemConfiguration.Framewor	libReachability.a 需要使用

播放器 SDK 依赖的开源库如表 6 表 5 所示。

表 6 SDK 需要的静态库

静态库名	含义
libCyberplayerUtils.a	CyberPlayerController 需要使用的工具库
libReachability.a	获取设备网络状态
libASIHTTP.a	网络通信库
libBase64.a	Base64 加密库
libGzipCompressor.a	Gzip 压缩算法库
libSBJSON.a	JSON 解析库

6.1.1 使用真机调试、发布

1. 配置 target，在 Xcode Build Phases->Link Binary With Libraries 中添加表 5 所列框架。
2. 如果你需要进行真机调试，从 Baidu-T5Player-SDK-iOS-1.1.zip 包中解压缩，在 Baidu-T5Player-SDK-iOS-1.1/lib/Release-iphonesos 目录下添加表 6 所列静态库，如果您的项目中已经使用了这些静态库，可以不再添加。
3. 将 Baidu-T5Player-SDK-iOS-1.1/lib/Release-iphonesos 目录下 libCyberPlayerController.a 和 libCyberPlayerController.h 添加到 Xcode 工程。

6.1.2 使用模拟器调试、发布

1. 配置 target，在 Xcode Build Phases->Link Binary With Libraries 中添加表 5 所列框架。
2. 如果你需要进行真机调试，从 Baidu-T5Player-SDK-iOS-1.1.zip 包中解压缩，在 Baidu-T5Player-SDK-iOS-1.1/lib/Release-iphonesos 目录下添加表 6 所列静态库，如果您的项目中已经使用了这些静态库，可以不再添加。
3. 将 Baidu-T5Player-SDK-iOS-1.1/lib/Release-iphonesimulator 目录下 libCyberPlayerController.a 和 libCyberPlayerController.h 添加到 Xcode 工程。

6.2 调用 API

下面介绍如何调用播放器 SDK 中已封装的 API 完成各项操作。

6.2.1 获取播放地址

如果您的应用在如下路径：/var/mobile/Applications/AA85D4F6-88dd-44F8-BFD6-E6D6B0B99069，您的应用 Documents 下有一个视频文件百度媒体云.rmvb。

以下代码将获得 Documents 路径：

```
NSArray* documentPaths =
NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask,
YES);
documentDir = [[documentPaths objectAtIndex:0] copy];
```

以下代码将获得播放百度媒体云.rmvb 文件的绝对路径：

```
NSString* videoFullPath = [documentDir stringByAppendingPathComponent:@"
百度媒体云.rmvb"]
```

此时 videoFullPath 的值为

/var/mobile/Applications/AA85D4F6-88dd-44F8-BFD6-E6D6B0B99069/Documents/百度媒体云.rmvb

6.2.2 传入 API key

```
NSString* msAK=@aaaaaaaaaaaaaaaaaaaaaaaaa";
NSString* msSK=@"bbbbbbbbbbbbbbbb";
[[CyberPlayerController class]setBAEAPIKey:msAK SecretKey:msSK];
```

其中 BAEAPIKey 需要传入全部值，secret key 只需要传入前 16 位，如果传入的 sk 长度大于 16 位则默认采取前 16 位。如果未传入 ak/sk，或者 ak/sk 错误，将导致播放行为失败并打印“Authentication fails!”。

6.2.3 初始化 CyberPlayerController

有多种方法完成 `CyberPlayerController` 的初始化

- 基本方式

```
CyberPlayerController *cbPlayerController = [[CyberPlayerController alloc] init];
```

当前 `CyberPlayerController` 1.1 版本及之前版本只支持单实例的 `CyberPlayerController` 对象，多实例将导致播放异常。

- 使用 `initWithContentURL`

```
NSURL *url = [NSURL URLWithString: videoFullPath];

if (!url)
{
    url = [NSURL URLWithString:[videoFullPath
stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding]];
}

CyberPlayerController *cbPlayerController = [[CyberPlayerController alloc] initWithContentURL:url];
```

此时 `initWithContentURL` 的参数值为: `file://localhost/var/mobile/Applications/`

`AA85D4F6-88dd-44F8-BFD6-E6D6B0B99069/Documents/%E7%99%BE%E5%BA%A6%E5%AA%92%E4%BD%93%E4%BA%91.rmvb`，可以看到将中文进行了 UTF8 编码。

- 使用 `initWithContentString`

```
CyberPlayerController *cbPlayerController = [[CyberPlayerController alloc] initWithContentString: videoFullPath];
```

调用 `initWithContentString` 时参数为

`/var/mobile/Applications/AA85D4F6-88dd-44F8-BFD6-E6D6B0B99069/Documents/百度媒体云.rmvb`，绝对路径中可以包含中文，播放器内部会完成字符串编码工作。

- 注意

当前 `CyberPlayerController` 1.1 版本及之前版本皆不支持百度影音资源（`bdhd://`开头的视频资源）。

6.2.4 设置播放地址

如果采用了基本方式初始化时，需要播放视频前需要设置播放文件地址。可以通过 `setContentURL` 或者 `setContentURL` 方法设置播放文件地址。

以下代码使用 `setContentURL` 设置本地播放路径，路径 `url` 的具体值为：

`file://localhost/var/mobile/Applications/`
`AA85D4F6-88dd-44F8-BFD6-E6D6B0B99069/Documents/%E7%99%BE%E5%BA%A6%E5%AA%92%E4%BD%93%E4%BA%91.rmvb`。

```
NSArray* documentPaths =
[NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask,
YES)];
documentDir = [[documentPaths objectAtIndex:0] copy];
NSString* videoFullPath = [documentDir stringByAppendingPathComponent:@"
百度媒体云.rmvb"]
CyberPlayerController *cbPlayerController = [[CyberPlayerController alloc]
init];
NSURL *url = [NSURL URLWithString: videoFullPath];
```

```
if (!url)
{
    url = [NSURLURLWithString:[videoFullPath
stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding]];
}

[cbPlayerController setContentURL: url];
```

以下代码使用 `setContentString` 设置网络播放路径，路径 `url` 的具体值为：`www.baidu.com/media/` 百度媒体云.rmvb(该地址为示例地址，无真实 `url`)。

```
NSString* videoFullPath = @"www.baidu.com/ media /百度媒体云.rmvb";
CyberPlayerController *cbPlayerController = [[CyberPlayerController alloc]
init];
[cbPlayerController setContentURL: videoFullPath];
```

当播放器正在播放视频时，设置 `contentString` 将不会导致播放新视频。如果希望播放新视频，需要调用 `prepareToPlay` 方法。

- 注意

当前 `CyberPlayerController` 1.1 版本及之前版本皆不支持百度影音资源（`bdhd://`开头的视频资源）。

6.2.5 播放视频

播放器在设置播放视频地址后，需要调用 `prepareToPlay` 方法对视频文件进行初始化工作。如果初始化完成后将发送 `CyberPlayerLoadDidPreparedNotification` 通知，并将 `isPreparedToPlay` 属性置为 `YES`。如果此时 `shouldAutoplay` 属性为 `YES`，则自动调用 `play` 方法进行播放。如果 `shouldAutoplay` 属性为 `NO`，则等待调用 `play` 或 `start` 方法播放。

- 基本方式

```
cbPlayerController.shouldAutoplay = YES
[cbPlayerController prepareToPlay];
```

此时，`shouldAutoplay` 属性为 `YES`，`prepareToPlay` 方法完成后将播放器自动调用 `play` 开始播放视频。

- 监听 `CyberPlayerLoadDidPreparedNotification` 通知

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(preparedDone:)
name:CyberPlayerLoadDidPreparedNotification
object:nil];
cbPlayerController.shouldAutoplay = NO
[cbPlayerController prepareToPlay];
```

同时新建 `preparedDone` 方法，并在该方法中调用 `start` 或 `play` 方法。

```
- (void) preparedDone: (NSNotification*) aNotification
{
    [cbPlayerController start];
}
```

- 注意

如果调用 `play` 方法前未调用 `prepareToPlay` 完成播放器对视频文件的初始化，则播放器自动调用 `prepareToPlay` 进行视频文件的初始化工作。

6.2.6 暂停播放

调用 `pause` 方法暂停视频播放，调用 `play` 或 `start` 方法重新开始播放。

```
[cbPlayerController pause];
```

6.2.7 改变播放位置

SDK 提供了多种改变播放位置的方式。

- 初始化时

当 `isPreparedToPlay` 属性为 NO 时,, 为了实现改变视频播放的初始时刻, 三种方式都可以达到相同效果: 设置 `initialPlaybackTime` 属性, 设置 `currentPlaybackTime` 属性, 调用 `seekto` 方法。

- 播放过程中

当播放视频过程中调用 `seekTo` 或者设置 `currentPlaybackTime` 属性, 将导致从指定位置开始播放。如果当前视频为网络视频, 可能触发缓冲 `CyberPlayerStartCachingNotification` 通知。如果希望在 UI 反应缓冲进度, 请监听 `CyberPlayerGotCachePercentNotification` 通知。

6.2.8 停止播放

```
[cbPlayerController stop];
```

该方法完成停止行为后, 将发送 `CyberPlayerPlaybackDidFinishNotification` 通知。

第7章 播放信息统计

媒体云服务为开发者提供了媒体应用播放的相关统计信息, 开发者可以通过查看统计信息了解应用的使用现状。

进入媒体云服务管理控制台步骤如下:

- 登录进入百度开发者中心的“管理中心”;
- 点击使用媒体云服务的应用, 进入应用基本信息页;
- 点击左侧边栏中的“云平台 > 媒体服务 > 媒体播放”, 即可进入播放器 SDK 统计查看页面。

第8章 联系我们

如果以上信息无法帮助您解决在开发中遇到的具体问题, 请通过以下方式联系我们:

邮箱: dev_support@baidu.com

百度工程师会在第一时间回复您。