蚂蚁科技

设备标识
使用指南

文档版本：20260302

蚂蚁集团
ANT GROUP

法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

蚂蚁集团 ANT GROUP及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.设备标识 ---- 05
1.1. 设备标识简介 ---- 05
1.2. 接入 Android ---- 05
1.3. 接入 iOS ---- 05
1.4. 接入 HarmonyOS NEXT ---- 07
1.4.1. 添加 SDK ---- 07
1.4.2. 使用 SDK ---- 07
1.5. 常见问题 ---- 07

1.设备标识

1.1. 设备标识简介

utdid 是一个 App 级别的设备标识 ID。通过设备标识组件，您可以简单快捷地获取设备 ID，以利于应用程序安全有效地找到特定设备。

> **说明**
> utdid 不能保证绝对的唯一性，存在重复的可能性。在对标识唯一性要求高的场景中不建议使用。

1.2. 接入 Android

设备标识支持原生 AAR 接入和组件化接入两种接入方式。

前置条件

- 若采用原生 AAR 方式接入，需先完成 将 mPaaS 添加到您的项目中 。
- 若采用组件化方式接入，需先完成 组件化接入流程。

添加 SDK

原生 AAR 方式

参考 AAR 组件管理，通过 **组件管理（AAR）** 在工程中安装 **设备标识符** 组件。

组件化方式

在 Portal 和 Bundle 工程中通过 **组件管理** 安装 **设备标识符** 组件。更多信息，参考 管理组件依赖。

获取设备标识

```
UTDevice
/**
 *  获取生成的设备唯一标识。
 *
 *
 *      @return 设备唯一标识。
 */
public static String getUtdid(Context context);
```

1.3. 接入 iOS

本文将介绍如何将设备标识组件接入到 iOS 客户端。设备标识支持 **基于 mPaaS 框架接入** 、**基于已有工程且使用 mPaaS 插件接入**以及 **基于已有工程且使用 CocoaPods 接入** 三种接入方式。您可以参考 接入方式介绍，根据实际业务情况选择合适的接入方式。

前置条件

您已经接入工程到 mPaaS。更多信息，请参见以下内容：

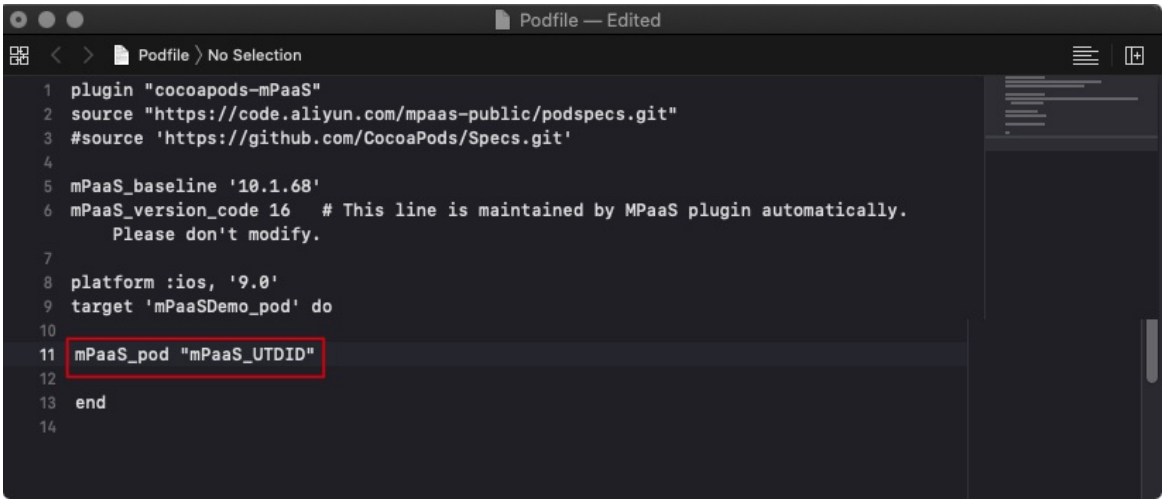
- 基于 mPaaS 框架接入
- 基于已有工程且使用 mPaaS 插件接入
- 基于已有工程且使用 CocoaPods 接入

添加 SDK

根据您采用的接入方式，请选择相应的添加方式。

- 使用 mPaaS Xcode Extension。此方式适用于采用了 **基于 mPaaS 框架接入** 或 **基于已有工程且使用 mPaaS 插件接入** 的接入方式。
  i. 点击 Xcode 菜单项 **Editor** > **mPaaS** > **编辑工程**，打开编辑工程页面。
  ii. 选择 **设备标识**，保存后点击 **开始编辑**，即可完成添加。

- 使用 cocoapods-mPaaS 插件。此方式适用于采用了 **基于已有工程且使用 CocoaPods 接入** 的接入方式。
  i. 在 Podfile 文件中，使用 `mPaaS_pod "mPaaS_UTDID"` 添加设备标识组件依赖。



  ii. 执行 `pod install` 即可完成接入。

使用 SDK

添加 SDK 后，您可以调用以下方法，获取设备的唯一标识：

```
/**
 获取 SDK 生成的设备唯一标识
 */
+ (NSString *)deviceId;
```

更多详情，请查看 `MPUTDIDAdapter.framework` 中的 `MPUtdidInterface.h` 文件。

1.4. 接入 HarmonyOS NEXT

1.4.1. 添加 SDK

本文介绍如何将设备标识组件接入到 HarmonyOS NEXT 客户端。您可以基于已有工程使用 ohpmrc 方式接入设备标识 SDK 到客户端。

前置条件

- 添加设备标识 SDK 前，请您确保已经将工程接入到 mPaaS。更多信息请参见开发指南 接入 mPaaS 能力。
- 完成 mppm 插件安装。详情请参考 安装 mppm 工具。

添加 SDK

通过 使用 mppm 工具 安装 **设备标识符** 组件。

1.4.2. 使用 SDK

初始化 SDK

初始化方案请参见开发指南 初始化 mPaaS。

获取设备标识

获取设备标识的方法如下：

```
import { MPFramework } from '@mpaas/framework'

let utdid = await MPFramework.instance.udid // 异步获取

let utdid = MPFramework.instance.udidSync // 同步获取
```

1.5. 常见问题

本文介绍 Android 系统中 utdid 相关的常见问题。

utdid 不变和变化的具体条件是什么？

- 手机系统或 App 重启
  - 无 SD Card 权限，**utdid 状态** 不变。

- 有 SD Card 权限，**utdid 状态** 不变。

- 删除 App 重新安装
  - 无 SD Card 权限，**utdid 状态** 变化。
  - 有 SD Card 权限，**utdid 状态** 不变。
- 设备双清，SD Card 数据被清除，重新安装
  - 无 SD Card 权限，**utdid 状态** 变化。
  - 有 SD Card 权限，**utdid 状态** 变化。

杀死进程后 utdid 发生了什么变化？

原因分析：

1. utdid 是应用软件 ID，每次启动时都会取应用外的 utdid 和应用内的 utdid 进行比对，并使用生成时间比较早的 ID。
2. 首次安装时，没有外存储权限，所以无法获得应用外的 utdid，而且应用内又不存在 utdid，于是会生成一个 utdid。
3. 当应用授权后，再次启动时，由于第 1 点中的机制，会用应用外的 utdid 覆盖应用内的 utdid，导致 utdid 发生变化。

解决方法：

1. 确保 utdid4all 的版本号不小于 1.1.5.3，因为旧版本与高版本的 Android 存在兼容问题。
2. 该问题只出现在首次安装应用时，当外存储权限变化时，再次启动 utdid 就不会变化了。

> **说明**
>
> Android 6.0 版本以后，部分手机需要在弹出外存储授权页，单击确定以后才能获得外存储权限，仅声明是无法获得的。

遇到 Android 10 以后的设备 ID 不变时如何处理？

随着设备权限越来越严格，utdid 已经很难保证设备统一、固定不变。如果遇到需要强调设备 ID 不变的场景，则需要使用其它获取 ID 的方案。